UNIVERSITY OF SÃO PAULO
SÃO CARLOS SCHOOL OF ENGINEERING

Lucas Daudt Franck

Design and Analysis of Divide-by-1.5/2 Prescalers

São Carlos

2024

Lucas Daudt Franck

Design and Analysis of Divide-by-1.5/2 Prescalers

Bachelor's Thesis presented to the Electrical Engineering Course of the São Carlos School of Engineering at the University of São Paulo in partial fulfillment of the requirements for the degree of Electrical Engineer.

Advisor: Prof. Dr. João Navarro Soares Jr.

São Carlos

2024

Catalog card prepared by Patron Service at "Prof. Dr. Sergio Rodrigues Fontes" Library at EESC/USP

Franck, Lucas Daudt
F822d Design and analysis of divide-by-1.5/2 prescalers / Lucas Daudt Franck ; Supervisor João Navarro Soares Junior. -- São Carlos, 2024.

Monograph Bachelor Final Thesis (Undergraduate in Electrical / Electronic Engineering) -- São Carlos School of Engineering, at University of São Paulo, 2024.

1. Dual-modulus prescaler. 2. Fractional frequency divider. 3. Double data rate. 4. High-speed digital circuit. I. Title.

Elena Luzia Palloni Gonçalves – CRB 8/4464

FOLHA DE APROVAÇÃO

Nome: Lucas Daudt Franck

Título: “Design and Analysis of Divide-by-1.5/2 Prescalers”

Trabalho de Conclusão de Curso defendido e aprovado em 24 / 06 / 2024 ,

com NOTA 10,0 (dez, zero), pela Comissão Julgadora:

Prof. Dr. João Navarro Soares Jr. - Orientador - SEL/EESC/USP

Prof. Dr. Maximiliam Luppe - SEL/EESC/USP

Prof. Assoc. João Paulo Pereira do Carmo - SEL/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Assoc. José Carlos de Melo Vieira Jr.

ABSTRACT

FRANCK, L. D. **Design and Analysis of Divide-by-1.5/2 Prescalers**. 2024. 65 p. Bachelor's Thesis (Course Conclusion Paper) - São Carlos School of Engineering, University of São Paulo, São Carlos, 2024.

Frequency synthesizers are fundamental components in modern wireless systems, providing the programmable frequency generation required for communication channel selection. Several design techniques are continuously developed to improve the key performance metrics of these blocks, which are mainly evaluated based on power consumption, maximum operating frequency, and phase noise. Fractional frequency dividers are an effective approach for reducing phase noise in fractional-N synthesizers by minimizing the feedback divider quantization error. Traditional fractional dividers, often based on phase-switching, phase interpolation, or double-edge triggering techniques, offer limited maximum operating frequency and high power consumption. Novel divide-by-1.5/2 prescaler circuits utilizing the little-explored double data rate (DDR) technique are proposed in this work and designed in the TSMC 65 nm CMOS process. The proposed DDR-based approach for half-integer step prescaler design combines aspects of phase-switching and double-edge triggering methods. Simulation results demonstrate that the DDR technique can increase the maximum operating frequency of the prescaler by up to 3.3x compared to conventional double-edge triggered flip-flop designs. Additionally, a novel balanced-DDR (B-DDR) version is introduced, which allows for 32% lower power consumption, achieves a 4x increase in maximum speed, and greatly mitigates the output period modulation observed in the simulations of other topologies.

Keywords: Dual-modulus prescaler. Fractional frequency divider. Double data rate. High-speed digital circuit.

RESUMO

FRANCK, L. D. **Design and Analysis of Divide-by-1.5/2 Prescalers**. 2024. 65 p. Bachelor's Thesis (Course Conclusion Paper) - São Carlos School of Engineering, University of São Paulo, São Carlos, 2024.

Os sintetizadores de frequência são componentes fundamentais dos modernos sistemas de comunicação sem fio, sendo responsáveis por gerar sinais com frequências configuráveis empregados na seleção dos canais de comunicação. Diversas técnicas são continuamente desenvolvidas para melhorar o desempenho desses blocos, que são avaliados primariamente quanto ao consumo de potência, à máxima frequência de operação e ao ruído de fase. Divisores de frequência fracionários são uma abordagem eficaz para minimizar o ruído de fase através da redução do erro de quantização do divisor no laço de realimentação. As implementações tradicionais de divisores fracionários são comumente baseadas em técnicas de comutação de fase, interpolação de fase ou em dispositivos sensíveis a dupla borda. Embora já validadas, essas técnicas apresentam frequência máxima de operação limitada e elevado consumo de energia. Nesse contexto, o presente trabalho apresenta novos circuitos divisores de frequência com módulo configurável de 1,5 ou 2 baseados na técnica de dupla taxa de dados (DDR) e projetados no processo CMOS de 65 nm da TSMC. Resultados de simulação demonstram que a técnica DDR pode aumentar a frequência máxima de operação do divisor em até 3,3x em comparação a um circuito equivalente implementado com flip-flops sensíveis a dupla borda. Adicionalmente, é apresentada uma nova abordagem de DDR balanceado, que permite uma redução de 32% no consumo de energia, um aumento de 4x na velocidade máxima de operação e uma redução significativa da modulação de período de saída verificada nas simulações das outras topologias.

Palavras-chave: Divisor de frequência fracionário. Dupla taxa de dados. Circuito digital de alta velocidade.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

B-DDR	Balanced-DDR
C²MOS	Clocked CMOS
CK	Clock
CKB	Complementary Clock Phase
CML	Current-Mode Logic
CMOS	Complementary Metal–Oxide–Semiconductor
DDR	Double Data Rate
DET	Double-Edge Triggering
DETFF	Double-Edge Triggered Flip-Flop
E-TSPC	Extended-TSPC
FSM	Finite State Machine
LPF	Low-Pass Filter
MC	Modulus Control
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
PD	Phase Detector
PDN	Pull-Down Network
PI	Phase Interpolation
PLL	Phase-Locked Loop
PS	Phase-Switching
PSD	Power Spectral Density
PSO	Particle Swarm Optimization
PUN	Pull-Up Network
RF	Radio Frequency
RFIC	Radio Frequency Integrated Circuit

RPS	Reversed Phase-Switching
SA	Simulated Annealing
SCL	Source-Coupled Logic
SETFF	Single-Edge Triggered Flip-Flop
SSCG	Spread Spectrum Clock Generator
TSMC	Taiwan Semiconductor Manufacturing Company
TSPC	True Single-Phase Clock
VCO	Voltage-Controlled Oscillator

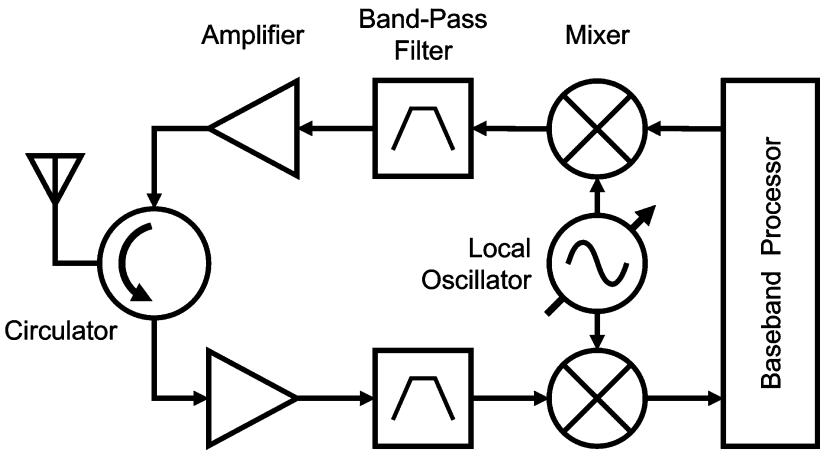
CONTENTS

1 INTRODUCTION

The ever-growing demand for speed and efficiency in modern communication systems drives the constant evolution of radio frequency (RF) circuits. Higher operating frequencies, novel modulation techniques, and increased channel density are a few of the strategies adopted to optimize the usage of the limited RF spectrum available. These improvements come as new challenges for RFIC engineers, who must design systems with better frequency response, higher selectivity, and lower noise, all while maintaining a small circuit area and minimal power consumption for mobile applications.

Oscillators are a fundamental building block of every RF system. These circuits generate the periodic waveforms required for frequency conversion, an operation that shifts a signal from one frequency range to another while maintaining its information. Figure 1 depicts a typical transceiver structure in which both the transmitter and receiver branches employ frequency conversion. Current transceiver architectures greatly benefit from frequency conversion as it allows for signal processing in the baseband while enabling the transmission and reception of data in the RF band, leading to more efficient utilization of the available RF spectrum [1, 2].

Figure 1 – Typical RF Transceiver.

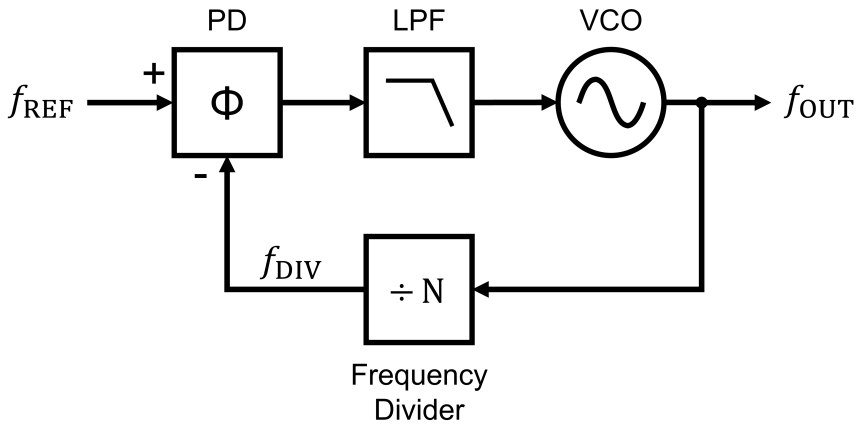


Source: based on RAZAVI (2011) [1].

Frequency conversion also enables RF transceivers to select various communication channels within a given frequency range. This feature, called channel selection, ensures reliable data transmission and reception without interference from other nearby channels [1]. Achieving efficient channel selection requires a variable local oscillator with a wide tunable range, high spectral purity, and great frequency stability. A commonly adopted approach to meet these conditions is using a programmable frequency synthesizer as the system local oscillator. While various frequency synthesizer architectures exist, the phase-locked loop (PLL) implementation is the prevalent one in high-performance transceivers [2].

Frequency synthesizers based on PLLs are popular in RF applications due to their high accuracy and flexibility. These systems typically consist of five main blocks: a reference frequency source that produces a signal with a stable frequency f_{REF}, a phase detector (PD), a low-pass filter (LPF), a voltage-controlled oscillator (VCO), and a programmable frequency divider. Figure 2 depicts an integer-N frequency synthesizer configuration [2]. By adjusting the division ratio of the programmable divider in the feedback network, the synthesizer generates an output frequency that is an N-multiple of the reference frequency. The system closed-loop configuration minimizes the phase error between the reference source and the feedback signal, allowing for a configurable and drift-free frequency synthesis [1].

Figure 2 – Integer-N Frequency Synthesizer.



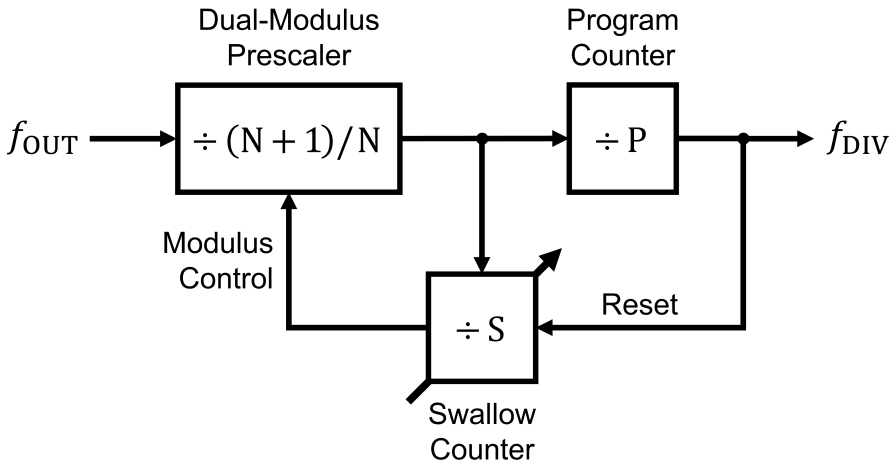
Source: based on LACAITA; LEVANTINO; SAMORI (2007) [2].

Integer-N synthesizers are named after the integer relationship between their output frequency and the reference frequency. While their simplicity has made them a popular choice, this architecture comes with a key design constraint: the reference frequency must either match or be an integer multiple of the channel spacing [1]. This limitation arises from the frequency dividers employed in these synthesizers, which are often implemented as modulo-N counters that are only capable of dividing the output frequency by integer values. As a result, the system output frequency step is restricted to be equal to the reference frequency, which may compromise the output signal noise and transient response, both of which depend on the values assumed by N [1, 2].

The pulse swallow divider is a typical implementation of the frequency divider found in integer-N synthesizers. This topology is slightly more complex than a standard digital counter and consists of three main blocks, as illustrated in Figure 3. The dual-modulus prescaler is a high-speed counter that provides a division ratio of either N or N+1, according to a modulus control input. The swallow counter is an auxiliary counter that divides its input frequency by a programmable factor S and generates the control

signal for the prescaler. Lastly, the program counter is a constant modulus divider that resets the swallow counter after P pulses are applied to its input. Among these blocks, the prescaler is the most critical as it receives its input signal directly from the VCO output. Consequently, it must operate at the same speed as the VCO with low power consumption and should have limited input capacitance to avoid affecting the VCO operation [1].

Figure 3 – Pulse Swallow Divider.



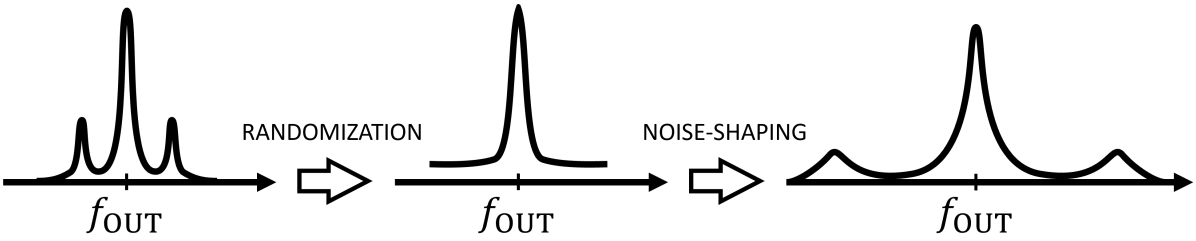
Source: based on RAZAVI (2011) [1].

A more sophisticated approach to signal generation that overcomes the integer divider limitation is the fractional-N frequency synthesizer. This synthesizer allows for a fractional relationship between its output frequency and the reference frequency, achieved by modifying the feedback divider. Instead of employing a digital counter to select the prescaler modulus, fractional-N synthesizers rely on a control pattern that dynamically toggles the prescaler counting modulo to achieve a non-integer averaged division ratio. For example, if a given prescaler spends half the time dividing by N and the other half by N+1, the time-averaged division ratio of this circuit equals N+0.5. This working principle enables greater freedom when choosing the reference frequency of the system, as it no longer needs to be an integer multiple of the channel spacing [1, 2].

Despite providing fine frequency resolution, fractional-N synthesizers face a key design challenge due to the fractional spurs introduced by the frequency divider operation. Since the instantaneous prescaler modulus is always an integer, either N or N + 1, the system is prone to building up phase error in the PD, leading to changes in the VCO tuning voltage [1]. For a system with a reference frequency f_{REF} and a prescaler controlled by a K-length pattern, the error signal at the PD output behaves like a periodic waveform that repeats every $f_{\mathrm{REF}} \times \mathrm{K}$ cycles. This phenomenon causes the modulation of the VCO, resulting in unwanted sidebands to appear in the frequency domain known as fractional spurs [2]. These spurs negatively impact the synthesizer phase noise and degrade the PLL output, requiring compensation [1, 3].

A popular strategy for suppressing fractional spurs is randomizing the prescaler control pattern and further applying noise-shaping techniques. The randomization breaks the periodic behavior of the loop and transforms the previously existing sidebands into phase noise. This newly introduced noise can then be shaped into a high-pass spectrum by employing a delta-sigma modulator to generate the divider pseudo-random control sequence [1]. Figure 4 provides a fairly simplified illustration of these steps and their effect on the frequency spectrum. The synthesizer natural low-pass response attenuates the high-pass-shaped noise and improves the system performance. To further reduce the noise, the overall PLL bandwidth should be lowered by adjusting the loop filter poles. Although suitable for some applications, this adjustment leads to slower response times and narrower stability margins [1, 4].

Figure 4 – Suppressing Fractional Spurs.



Source: based on RAZAVI (1998) [5].

An additional approach to further suppress phase noise in fractional-N synthesizers is by decreasing the division step size of the prescaler. To fully understand the impacts of such a change, it is important to establish a mathematical relationship between the average power of the output-referred phase noise and the prescaler quantization step. The following paragraphs will present the complete deduction of this relationship, starting from the basics and explaining every non-trivial step [1, 6, 7].

The first core concept worth revisiting is the inherent relationship between phase and frequency. Equation 1.1 starts from the definition of angular frequency ω (in radians per second) and shows that the phase ϕ (in radians) of a given signal can be calculated by integrating its frequency f (in Hertz) over time. Equation 1.2 illustrates this same relationship in the frequency domain, which is obtained by taking the Fourier Transform of the previous expression. Here, $\Phi(f)$ and $F(f)$ are the frequency-domain representations of the phase function $\phi(t)$ and the frequency function $f(t)$, respectively. Finally, Equation 1.3 presents the phase power spectral density expression, where $S(f)$ represents the power spectral density (PSD) of a signal.

$$f(t) = \frac{\omega(t)}{2\pi} = \frac{1}{2\pi} \cdot \frac{d\phi(t)}{dt} \;\; \therefore \;\; \phi(t) = 2\pi \int_0^t f(\tau) d\tau \tag{1.1}$$

$$\mathfrak{F}\{\phi(t)\} = \Phi(f) = 2\pi \cdot \frac{1}{j2\pi f} F(f) = \frac{1}{jf} F(f) \tag{1.2}$$

$$S_{\Phi}(f) = \left|\frac{1}{jf}\right|^2 S_F(f) = \frac{1}{f^2} S_F(f) \tag{1.3}$$

After recapping the above expressions, the next step is to determine the frequency error at the divider output relative to its quantization step. Assuming that the prescaler is controlled by a random binary sequence $b(t)$, Equation 1.4 defines the instantaneous frequency at the divider output $f_{\mathrm{DIV}}(t)$ in terms of its counting modulo N and the synthesizer output frequency $f_{\mathrm{OUT}}(t)$. Since $b(t)$ is a random variable with a nonzero mean, it can be expressed as its average value α plus a zero-mean random quantization error $q(t)$. It is worth noting that N + α equals the average division ratio of the feedback divider.

$$f_{\mathrm{DIV}}(t) = \frac{f_{\mathrm{OUT}}(t)}{N + b(t)} = \frac{f_{\mathrm{OUT}}(t)}{N + \alpha + q(t)} \tag{1.4}$$

In an ideal phase-locked system, the frequency error $f_{\mathrm{ERR}}(t)$ can be expressed as the difference between the reference frequency f_{REF} and $f_{\mathrm{DIV}}(t)$, as shown in Equation 1.5. Further simplification can be achieved by taking the linear approximation via Taylor Expansion of the term inside the brackets and realizing that the output frequency divided by the average division ratio equals f_{REF}. This approximation leads to Equation 1.6, which is sufficiently accurate for applications where $q(t) << \mathrm{N} + \alpha$.

$$f_{\mathrm{ERR}}(t) = f_{\mathrm{REF}} - \frac{f_{\mathrm{OUT}}(t)}{N + \alpha + q(t)} = f_{\mathrm{REF}} - \frac{f_{\mathrm{OUT}}(t)}{N + \alpha}\left[1 + \frac{q(t)}{N + \alpha}\right]^{-1} \tag{1.5}$$

$$f_{\mathrm{ERR}}(t) \approx f_{\mathrm{REF}} - f_{\mathrm{REF}}\left[1 - \frac{q(t)}{N + \alpha}\right] \approx f_{\mathrm{REF}} \cdot \frac{q(t)}{N + \alpha} \tag{1.6}$$

Since the quantization error $q(t)$ is inherently a random signal, treating the frequency error as a deterministic signal makes no sense. Instead, considering the error as random noise is more appropriate as it allows for a more meaningful stochastic analysis. Accordingly, the PSD of the frequency noise at the divider output can be expressed by Equation 1.7. Next, the frequency noise referred to the synthesizer output is obtained by multiplying Equation 1.7 by the average division ratio squared, as shown in Equation 1.8. This operation is akin to the multiplier effect of the feedback divider on the reference frequency.

$$S_{f_N}(f) \equiv S_{f_{\mathrm{ERR}}}(f) = \left(\frac{f_{\mathrm{REF}}}{N + \alpha}\right)^2 \cdot S_q(f) \tag{1.7}$$

$$S_{f_{N\,\mathrm{OUT}}}(f) = (N + \alpha)^2 \cdot S_{f_N}(f) = f_{\mathrm{REF}}^2 \cdot S_q(f) \tag{1.8}$$

The output-referred phase noise with respect to the quantization noise PSD is assessed by combining Equation 1.3 and Equation 1.8, as shown in Equation 1.9. The missing part needed to establish a relationship between the synthesizer phase noise and the prescaler step size δ is finding the quantization noise PSD. Knowing that the average division ratio α varies within the $0 \leq \alpha \leq \delta$ range, the instantaneous quantization error may

assume one of two possible values: $(\delta - \alpha)$ when the prescaler is set to the N + δ modulus, or $-\alpha$ when configured in the divide-by-N mode. Notice that for $q(t)$ to reach a zero mean, the value $(\delta - \alpha)$ must appear with probability α/δ, and the value $-\alpha$ must appear with probability $(\delta - \alpha)/\delta$. For a binary control sequence $b(t)$ consisting of square pulses of width T_b that repeat randomly at a rate of $1/T_b$, the stochastic analysis presented in [1] proves that the quantization noise PSD can be expressed by Equation 1.10.

$$S_{\Phi_{N\,\mathrm{OUT}}}(f) = \frac{1}{f^2}\, S_{f_{N\,\mathrm{OUT}}}(f) = \left(\frac{f_{\mathrm{REF}}}{f}\right)^2 \cdot S_q(f) \tag{1.9}$$

$$S_q(f) = \frac{\alpha(\delta - \alpha)}{T_b} \left[\frac{\sin(\pi T_b f)}{\pi f}\right]^2 \tag{1.10}$$

A quick examination of Equation 1.10 reveals that the frequency-independent term $\alpha(\delta - \alpha)$ varies between zero, whenever α assumes 0 or δ, and its maximum value of $\delta^2/4$, when α equals $\delta/2$. Thus, the worst quantization noise PSD case is given by Equation 1.11, which is the final step in establishing a straightforward mathematical relationship for the output phase noise in terms of the prescaler step size δ. From this expression, the dependence of the phase noise PSD on $\delta^{\,2}$ can be verified. Consequently, if δ is reduced by half, the PSD decreases by a factor of 4, which corresponds to a 6 dB noise reduction.

$$S_{q\,max}(f) = \frac{\delta^{\,2}}{4T_b} \left[\frac{\sin(\pi T_b f)}{\pi f}\right]^2 \tag{1.11}$$

Numerous implementations of fractional frequency dividers with various step sizes have been proposed. Most circuits typically follow one of three main design approaches: phase-switching (PS), phase interpolation (PI), or double-edge triggering (DET). The PS and PI techniques can achieve very small quantization steps but require quadrature signal generation and complex phase selection circuitry, which limits their maximum operating speed. On the other hand, DET-based designs are much simpler but offer limited division ratios and have higher power consumption. Hence, fractional frequency dividers for applications requiring high-speed operation and low power consumption present serious design challenges.

This work proposes new half-integer step dual-modulus prescaler topologies based on the double data rate (DDR) structures introduced in [8] and [9]. This novel design approach combines aspects of traditional PS and DET methods, enabling high-speed circuits with lower power consumption. Conventional flip-flops were employed in the new counter circuitry, while phase-switching logic was used for merging two state variables at the output. Three divide-by-1.5/2 prescaler circuits were implemented using DDR and tested against an equivalent DET design. For all implementations, the transistor sizing and the circuit optimization were performed using a metaheuristics framework to avoid biased comparisons. The design steps presented in this work can be replicated to create different digital circuits.

1.1 Aims and Objectives

This work aims to comprehensively review the main techniques used in the design of fractional frequency dividers, examining their strengths and weaknesses, and propose a new approach better suited for high-speed, low-power applications. The main objective is to adapt the existing DDR technique to design a half-integer step dual-modulus prescaler and compare its performance against an equivalent DET implementation.

1.2 Organization

This work is organized into five chapters: Introduction, Literature Review, Divide-by-1.5/2 Prescaler Design, Results and Discussion, and Conclusion. The **Introduction** contextualizes the research topic and presents some key design challenges and techniques for lowering phase noise in frequency synthesizers. The **Literature Review** reviews the leading publications on fractional frequency dividers and summarizes the three most popular design approaches. The **Divide-by-1.5/2 Prescaler Design** explains the methods and procedures carried out throughout the project to achieve the research objectives. The **Results and Discussion** presents extensive simulation data and evaluates the performance of the proposed DDR technique. Finally, the **Conclusion** highlights the most important results and suggests topics for further research.

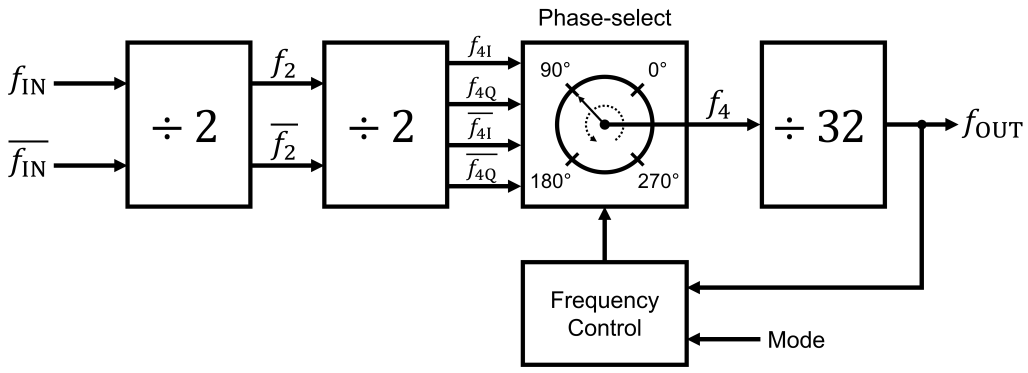
2 LITERATURE REVIEW

The following chapter presents the leading publications on fractional frequency dividers in the context of high-performance frequency synthesizers. The articles cited here represent a sample of the most relevant publications on the topic but by no means exhaust the entire literature. Since these dividers are synthesizer subblocks, some publications introduce the new prescaler topologies as part of complete systems, often without detailed explanations or individual performance metrics. Consequently, the level of detail in the following descriptions may vary based on the emphasis given to the divider block in the source publication. The three most popular design techniques will be explored individually, and a complete comparison will be provided at the end of the chapter, summarizing the key characteristics of each approach.

2.1 Phase-Switching

The phase-switching technique for frequency dividers was initially proposed by Craninckx and Steyaert (1996) [10] as an asynchronous alternative to counter-based prescalers. The operation of phase-switching dividers relies on multiplexing different phases of the same signal to assemble an output waveform with the desired timing characteristics. For example, if an output signal is about to transition from low to high, but its 90° lagging phase is selected as the new output, the transition will be delayed by a quarter period. This class of dividers exhibits lower power consumption due to its asynchronous nature but is more susceptible to glitches because of constant phase commutations. Figure 5 illustrates the diagram of the divider proposed in the original 1996 paper.

Figure 5 – Phase-Switching Divider.



Source: based on CRANINCKX; STEYAERT (1996) [10].

Wang *et al.* (2009) [11] presented a half-integer step multi-modulus programmable divider that combined reversed phase-switching (RPS) with a zipper divider structure. This approach enhanced design flexibility and achieved significant power savings, reducing overall consumption by 21% on average. The proposed circuit utilizes a 4-to-1 multiplexer to switch between four 90°-spaced phase signals in a decreasing sequence (hence the name RPS), which enables glitch-free operation. The final design was implemented using source-coupled logic and simulated in the Chartered Semiconductor 180 nm CMOS process. However, no simulation data regarding the maximum operating frequency or absolute power consumption was provided in the paper.

Jin *et al.* (2012) [3] proposed a glitch-free phase-switching divide-by-0.5/1/1.5/2 cell with reduced power consumption and higher operating frequency. Their circuit was implemented in a 180 nm CMOS process and fully tested, showing a power consumption of 9 mW at a 2 GHz input frequency. The design ensures glitch-free operation across all supported division ratios at the cost of increased circuit complexity. The divider cell consists of a divide-by-2 block with a 4-phase quadrature output, a multiplexer based phase selector, and a digital controller, all of which must be carefully designed to guarantee correct and reliable operation.

Thirunarayanan *et al.* (2013) [12] introduced a novel 0.2 step-size programmable fractional frequency divider based on 5-phase switching. The divider comprises a 5-stage injection-locked ring oscillator, a finite state machine, a dynamic divider chain, a phase combiner, and some resynchronization circuitry. The different phases generated by the ring oscillator are linearly combined in the phase combiner, following a sequence dictated by the finite state machine. The divider was implemented in 65 nm CMOS technology and consumes 850 µA at a 1.1 V supply when operating at 2.56 GHz. The circuit was primarily designed with only digital cells, which greatly benefits from scaling technology.

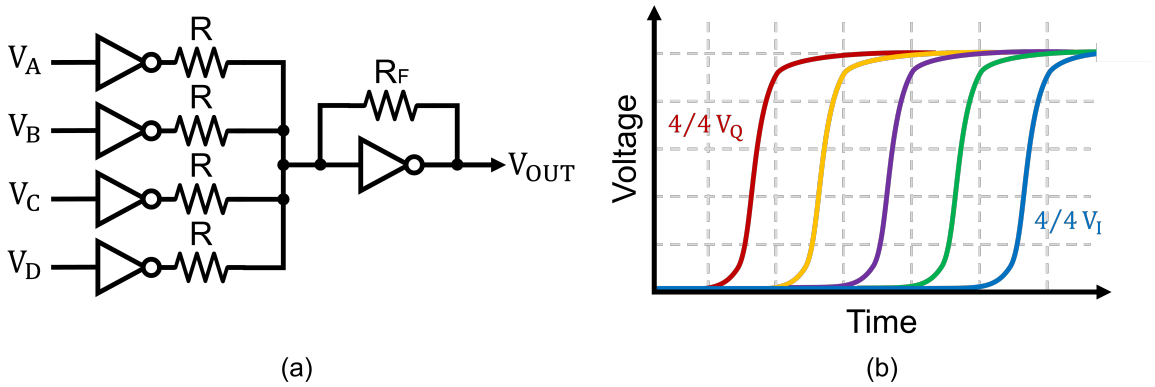
Hu *et al.* (2020) [13] presented a 0.5-step phase-switching multi-modulus frequency divider as part of a 0.045 to 2.5 GHz wideband frequency synthesizer. The divider operates within the 2.5 to 5.0 GHz range and was implemented using the TSMC 180 nm RF CMOS process. It employs a forward-switching logic that selects the output signal from a 4-phase quadrature generator in a positive sequence, offering great driving capability at high frequencies but being prone to glitches. The entire frequency synthesizer consumes a total power of 108 mW, but no specific data regarding the power consumption of the divider alone was provided.

2.2 Phase Interpolation

Phase interpolation is a popular technique used in RF systems to generate signals with controlled phase shifts. This method involves combining two or more input signals to produce an output signal whose phase is equivalent to the weighted sum of the input

phases. In fractional frequency dividers, phase interpolation allows for precise control over signal transitions, enabling programmable delay lines that modify the timing behavior of the signal. Figure 6(a) illustrates a classical quadrature-based phase interpolator circuit with four inputs. Figure 6(b) shows all possible output signals based on the number of input signals that are in-phase (V_I) or in quadrature (V_Q). For example, if all four input signals are V_I, the output behaves like the right-most signal in blue (4/4 V_I); if any three inputs are V_Q and the remaining one is V_I, the output transition is equivalent to the yellow curve (3/4 V_Q + 1/4 V_I) [14].

Figure 6 – Phase Interpolation Example.



Source: based on RAZAVI (2023) [14].

Tasca *et al.* (2011) [15] presented a novel delta-sigma fractional-N digital PLL based on a single-bit time-to-digital converter that features a truly fractional frequency divider. The PLL was implemented in a standard 65 nm CMOS process and achieves a tuning range from 2.92 GHz to 4.05 GHz while maintaining a total power consumption of 4.5 mW. The truly fractional divider was realized as an integer divider followed by a digitally controlled delay stage, which achieves a 10-bit resolution. A phase-interpolator N/(N+0.5) is used in the divider to limit the maximum time variation at the output due to the number of quantization levels of the delta-sigma modulation. The paper focused mainly on the complete system; therefore, no simulation or measurement data about the individual divider performance was available.

Liu *et al.* (2014) [16] proposed a phase interpolation-based half-integer frequency divider for use in a Spread Spectrum Clock Generator (SSCG). The fractional frequency divider was realized by combining an integer multi-modulus frequency divider with a resynchronization block to mitigate jitter accumulation throughout the divider chain. Additionally, a phase interpolation circuit at the end of the divider chain enabled half-integer step division by extracting 180-degree phase information from the feedback signal. The SSCG was implemented in 180 nm CMOS technology and successfully operates at 3 GHz with a power consumption of 12 mW under a 1.8 V power supply. No test data regarding the fractional frequency divider was provided.

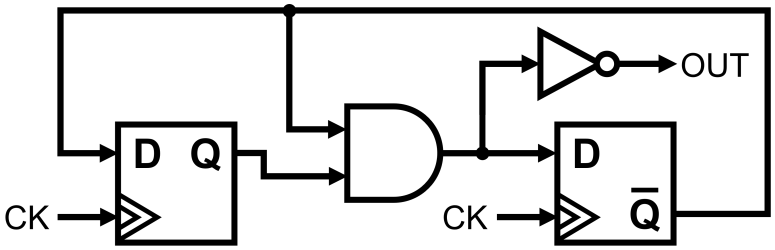
Lin *et al.* (2017) [17] presented a fractional-N frequency divider based on a 5-bit phase interpolator implemented in 65 nm CMOS technology. The design features a novel phase-interpolator unit with a switched resistor load and an all-digital phase rotating scheme that operates with quadrature input signals. Phase interpolation is effectively achieved by the weighted current summation of the two quadrature input signals on the resistive load. The divider circuit was simulated at 2.4 GHz and consumes 1.3 mW under a 1.2 V supply. No data on the maximum operating frequency was provided.

Xu *et al.* (2022) [18] introduced a clock generator that operates in the 0.73-15.5 GHz frequency range implemented in a 12 nm FinFET CMOS process. The design features a unique divide-by-1.5 divider to minimize the required VCO tuning range, which is realized through duty cycle interpolation between two digital divide-by-1.5 frequency dividers with 33.3% and 66.7% output duty cycles. The divider leverages delay chains that average the positions of rising and falling edges, making it susceptible to even-cycle-to-odd-cycle period errors due to imbalances in the pulse combination circuitry. The complete clock generator consumes 19.5 mW at a 13.28 GHz output under a 0.9 V supply, while the fractional divider consumes 4.1 mW under the same conditions.

2.3 Double-Edge Triggering

Double-edge triggering is a common technique in high-speed digital circuits that enables doubling throughput without increasing the clock frequency. Its operating principle is straightforward and lies in triggering events on both the rising and falling edges of the clock signal. Fractional frequency dividers based on double-edge triggered flip-flops (DETFFs) are popular due to their simplicity and reliability, as they can be designed similarly to conventional synchronous counters [1]. For example, consider a digital counter designed with conventional single-edge triggered flip-flops (SETFFs) that has a counting modulo of five. Under normal operating conditions, this circuit would take five clock cycles to start over. However, by replacing all SETFFs in this same counter circuit with DETFFs, the sequence would only take 2.5 clock periods to loop over thanks to state transitions occurring on both the positive and negative edges of the clock. Figure 7 shows a simple divide-by-1.5 counter circuit based on this working principle.

Figure 7 – Divide-by-1.5 Circuit.



Source: based on RAZAVI (2011) [1].

Huang *et al.* (2004) [19] presented a divide-by-64/64.5 prescaler using double-edge-triggered D-flip-flops. The design was implemented in a 250 nm CMOS process and dissipated 5.525 mW of power while operating at a 1.8 GHz clock. It employs dynamic D-flip-flops in the divider circuitry to shorten setup and hold times, enabling higher operating frequencies. The design was based on a synchronous divide-by-2/2.5 cell followed by an asynchronous divide-by-32 counter. Simulation results highlighted a 35% power consumption reduction when compared to a conventional divider for a similar application.

Moon *et al.* (2005) [20] proposed a novel divide-by-16.5 frequency divider tailored for Ethernet applications. The design cascades a divide-by-3 stage with a divide-by-5.5 cell to achieve the desired division ratio. The divide-by-5.5 cell was based on double-edge-triggered flip-flops, which were built using two level-sensitive latches and a multiplexer. The divider was implemented using current-mode logic and fabricated in 130 nm CMOS technology. Experimental measurements showed that the frequency divider could operate at frequencies exceeding 5 GHz while maintaining a power consumption of 18 mW from a 1.2 V power supply.

Yang *et al.* (2006) [4] introduced a delta-sigma fractional-N frequency synthesizer featuring a half-step frequency divider that reduces quantization noise by 6 dB. The feedback divider is based on a 1/1.5 divider cell, which employs two complementary level latches and a multiplexer to form a double-edge-triggered flip-flop. The complete synthesizer chip was fabricated using a 180 nm CMOS process and consumes 47.8 mW of total power while operating in the 2.5-3.2 GHz range. Post-layout simulations demonstrate that the 1/1.5 divider cell can effectively operate at frequencies exceeding 4.6 GHz.

Lu *et al.* (2009) [21] presented a wideband fractional-N synthesizer with adaptive bandwidth control fabricated using a 180 nm CMOS process. The system employs a 4/4.5 prescaler based on four flip-flops, two multiplexers, and two D-latches to reduce quantization noise. The combined operation of the prescaler blocks forms a double-edge triggered counter. The complete synthesizer consumes 25 mW under a 1.8 V power supply. The paper provides no significant information regarding the fractional divider, as it focuses more on details about the frequency synthesizer.

2.4 Techniques Comparison

Each of the three techniques presented above has unique characteristics that make them more or less suitable for certain applications. There is no overall best approach, but for fractional divider applications, the techniques can be objectively evaluated based on the following three main metrics: quantization step, operating frequency, and power consumption. Table 1 summarizes these metrics for each of the previously listed designs. The operating frequency lists the highest working frequency directly or indirectly stated in the original publication and the power consumption is expressed in mW/GHz to

facilitate comparison. It is important to note that some designs comprise complete dividers, while others are only prescalers. Therefore, direct comparison between all items is not as straightforward as it may seem.

Table 1 – Fractional Dividers Comparison.

Work	**Year**	**Node (nm)**	**Technique**	**Quantization Step**	**Frequency (GHz)**	**Power (mW/GHz)**
[11]	2009	180	PS	0.50	3.00	-
[3]	2012	180	PS	0.50	2.00	4.50
[12]	2013	65	PS	0.20	2.56	0.37
[13]	2020	180	PS	0.50	5.00	12.08
[15]	2011	65	PI	0.50	4.05	-
[16]	2014	180	PI	0.50	3.00	-
[17]	2017	65	PI	0.03	2.40	0.54
[18]	2022	12	PI	0.20	15.50	0.31
[19]	2004	250	DET	0.50	1.80	3.07
[20]	2005	130	DET	0.50	5.00	3.49
[4]	2006	180	DET	0.50	4.60	7.72
[21]	2009	180	DET	0.50	1.96	-

Classifying various circuits based solely on performance metrics does not provide a comprehensive understanding, as different manufacturing processes and measurement procedures significantly impact the numerical values obtained. Fractional divider techniques can also be categorized by their complexity. Designs based on phase switching often require polyphase signal generation and careful control over the switching sequence to avoid glitches, making them high-complexity systems. Alternatively, phase interpolation circuits need quadrature signal generation and a balanced interpolator, ranking them as medium-complexity since quadrature signals are more easily obtained than polyphase ones. Finally, dividers based on double-edge triggering are much simpler and are often built with digital circuits that behave like double-edge triggered flip-flops, categorizing them as low-complexity designs.

3 DIVIDE-BY-1.5/2 PRESCALER DESIGN

The activities performed within the design of the divide-by-1.5/2 prescalers are outlined in the following chapter. A prescaler division ratio of 1.5/2 was selected to evaluate the feasibility of adapting the double data rate (DDR) structures presented in [8] and [9] for implementing fractional divider circuits. The 1.5/2 division ratio is the smallest useful ratio for RF applications, as a 1/1.5 circuit wastes power by not performing any frequency division while operating with modulus one. The performance of the newly adapted DDR technique was compared against equivalent DET prescaler circuits. All evaluated designs underwent automated transistor sizing and circuit optimization performed by a metaheuristics framework to avoid any design bias.

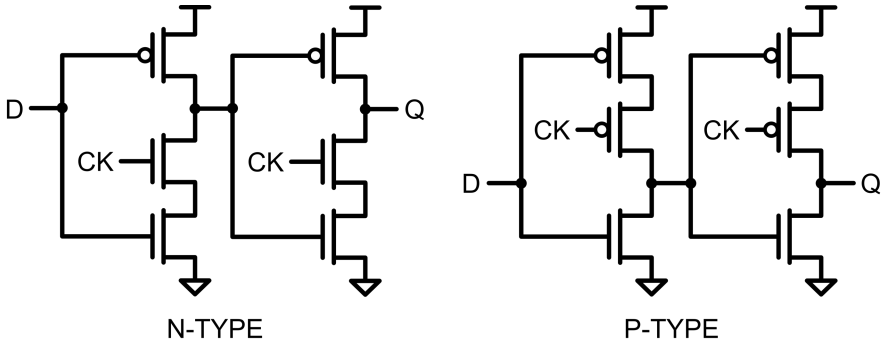
3.1 High-Speed Digital Design

High-speed digital design is a subfield of RFIC design that combines both classical digital logic and RF design principles. Among many techniques, the two most popular logic styles employed to achieve high-performance digital circuits are Current-Mode Logic (CML), also known as Source-Coupled Logic (SCL), and True Single-Phase Clock (TSPC) [1]. CML affords the fastest circuits and is structured around differential pairs. It requires differential inputs and produces differential outputs, making it the most adopted logic for balanced applications. On the other hand, TSPC logic offers lower power dissipation and rail-to-rail output swings, making it suitable for interfacing with standard CMOS digital logic voltage levels [1]. The DDR structures from [8] and [9] were based on TSPC, and therefore TSPC was the selected logic style in this work.

3.1.1 True Single-Phase Clock

The TSPC logic was first introduced by Yuan, Karlsson, and Svensson in 1989 [22] as a single-phase logic alternative to the previous existing clocked CMOS (C²MOS) [23], Domino Logic [24], and NORA Logic [25] structures. The key idea presented by the authors was that single-phase data storage units with embedded logic could be built by cascading precharge stages and clocked logic stages (often called static stages) [22]. While this definition is a big simplification of the complete TSPC technique, it offers a starting point for developing some intuition about the topic. Some further understanding can be obtained by analyzing a simple TSPC data latch, which is built by connecting double N-type or P-type clocked inverters in series, as shown in Figure 8 [26, 27]. This circuit is an example of a series combination of two static stages.

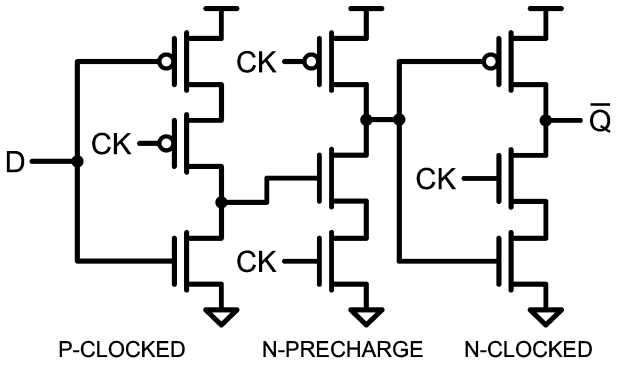
Figure 8 – TSPC Latches.



Source: based on YUAN; SVENSSON (1989) [26].

Taking the N-type latch as an example, it is clear that when the clock (CK) signal is HIGH, the circuit behaves as two inverters in series, copying any input value to the output. When the clock signal goes LOW, the output nodes of the clocked inverters cannot change to LOW, and thus the previous data is retained at the latch output. Therefore, the circuit behaves as a single-phase clock latch. A similar analysis can be applied to the P-type version. A simple TSPC flip-flop can be created by combining the N-type latch in series with the P-type latch. However, this implementation is not the most reliable one, as it is more susceptible to race conditions – both latches may become transparent simultaneously during the clock transition [28]. A more robust and efficient TSPC flip-flop that employs both precharge and clocked logic stages is shown in Figure 9.

Figure 9 – TSPC Flip-Flop.



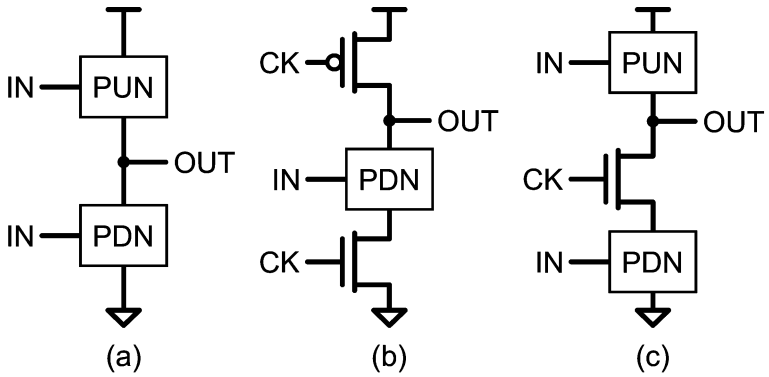
Source: based on NAVARRO (1998) [29].

The TSPC flip-flop shown in Figure 9 consists of two blocks: a P-type pseudo-latch and an N-type precharge latch. The pseudo-latch is formed by a P-type clocked inverter that, when the clock signal is HIGH, only allows the output node to transition from HIGH to LOW – hence its status, as a true latch would maintain the output regardless of the input. The second block is an N-type precharge latch, formed by cascading an N-type precharge stage with an N-type clocked inverter. When the clock signal is LOW, the input

of the N-type clocked inverter is precharged to HIGH, and its output retains the last logical value by remaining in a high-impedance state. Once the clock goes HIGH, the latch enters the evaluation phase, during which its output initially follows the input of the precharge stage. However, this N-type latch has an additional and important feature: during the evaluation phase, any input change from HIGH to LOW does not affect the latch output. Therefore, the combined operation of the P-type pseudo-latch and the N-type precharge latch compensates for the fact that the pseudo-latch is not a true latch, yielding a smaller and more robust TSPC flip-flop.

The previous flip-flop example illustrates a practical approach to understanding the operation of TSPC circuits. This same idea of analyzing individual and combined stages is key to breaking down complex TSPC blocks into simpler circuits. Another core feature of the TSPC style is that logic gates can be embedded directly into the basic structures, as depicted in Figure 10 [28]. Figure 10(a) shows a generic static CMOS logic gate comprised of a Pull-Up Network (PUN) and a Pull-Down Network (PDN). Figure 10(b) and Figure 10(c) illustrate how these networks can be integrated into N-type precharge stages and N-type clocked logic stages, respectively [29].

Figure 10 – Embedding Logic into TSPC Stages.



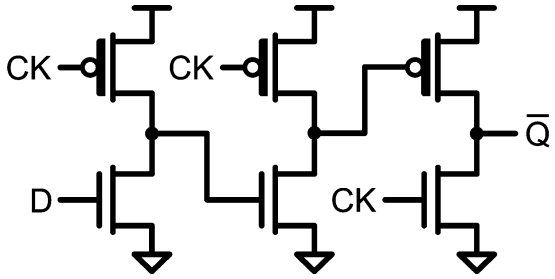
Source: based on NAVARRO (1998) [29].

In summary, the TSPC technique introduces a number of basic logic structures and a set of rules for connecting them to build reliable high-speed digital circuits. TSPC is compatible with both static and dynamic logic styles and is suited for single-phase small-area and low-power applications. Despite not matching the speed of CML, TSPC often offers more than enough speed performance with much better energy efficiency [1]. Designing compact TSPC circuits is not a trivial task and requires a good understanding of the technique's principles, which were hopefully introduced at a basic level in this section. By no means was the TSPC technique fully covered here, and further reading of the original publications is highly encouraged [22, 26–29].

3.1.2 Extended True Single-Phase Clock

The Extended-TSPC (E-TSPC) technique is a superset of TSPC design rules proposed by Navarro and Van Noije in 1999 [8]. The E-TSPC extends the TSPC technique by introducing new ratioed structures and additional composition rules, enabling faster circuits with lower power consumption and smaller areas. These additional rules also help avoid race conditions and preserve data during the holding phases [30]. Figure 11 shows an E-TSPC-based flip-flop circuit, whose operation is quite similar to the TSPC version presented in Figure 9 but using ratioed structures. In the following schematic, a thick gate MOSFET represents a stronger device with higher capability to conduct current.

Figure 11 – E-TSPC Flip-Flop.

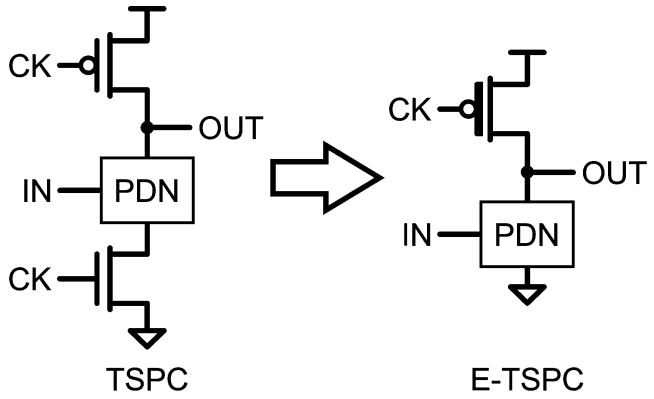


Source: based on NAVARRO (1998) [29].

Ratioed logic structures allow for faster circuit designs but come at the cost of higher static power consumption. This increase in static power stems from the basic operating principle of ratioed blocks, where two or more MOSFETs may fight to impress the output logic level, creating a low-impedance path between the source and ground. While this seems like a significant drawback, in very high-speed CMOS circuits, dynamic power is the main source of energy dissipation. The intense transistor switching activity leads to successive charge and discharge cycles of MOSFET gates and parasitic capacitances, causing substantial energy loss. Hence, for high-speed applications, static power consumption becomes much less important, and using ratioed structures may yield better energy efficiency due to their reduced transistor count [8, 28].

Similar to TSPC primitives, E-TSPC ratioed blocks also support embedding digital logic within their structure. Figure 12 illustrates how to transform a generic TSPC N-type precharge stage into its E-TSPC equivalent, often referred to as an NMOS-like structure due to its resemblance to older NMOS logic [8]. As in the previous schematic, the thick gate MOSFET represents a stronger device with higher current capability. A brief circuit analysis of both structures clearly shows their logic equivalence, but the E-TSPC version creates a low-impedance path between source and ground when the clock (CK) is LOW and the input signal (IN) is HIGH [29].

Figure 12 – Embedding Logic into E-TSPC Stages.



Source: based on NAVARRO (1998) [29].

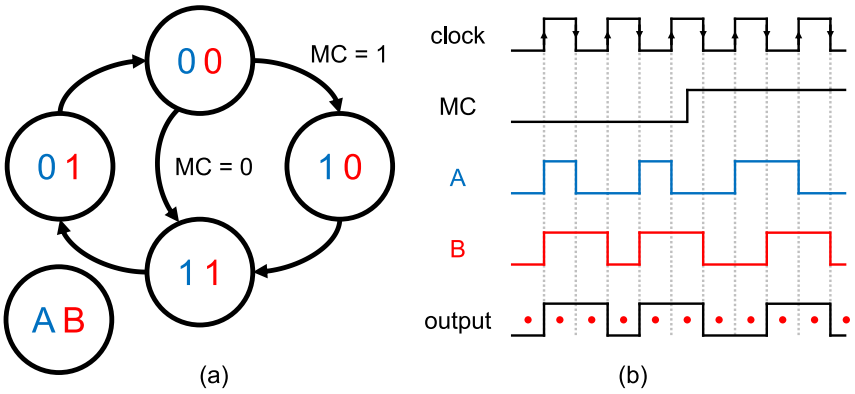
The logical equivalence between TSPC and E-TSPC structures makes them interchangeable in circuit designs. This feature is extremely useful when trying to improve circuit performance, as a TSPC speed bottleneck may be solved by swapping the critical stage with an E-TSPC equivalent structure. Similarly, a power-hungry E-TSPC circuit may have its power consumption reduced by replacing its slowest stage with a TSPC version. It is important to note that NMOS-like E-TSPC structures require careful transistor sizing to properly operate at high speeds with medium to low power consumption.

3.2 DET Technique

Fractional frequency dividers based on DET are quite flexible and straightforward to design. As previously seen in the Literature Review, the most popular approach for designing DET dividers is utilizing DETFFs or digital circuits with equivalent behavior. A divide-by-1.5/2 prescaler can be easily implemented by building a standard 3/4-modulus counter with SETFFs, and then swapping all SETFFs for DETFFs. Because the counter value is incremented on both rising and falling clock edges due to DETFFs, the system requires half the clock transitions to return to the count beginning. Therefore, the practical implication of replacing SETFFs with DETFFs in a digital counter is halving its modulus.

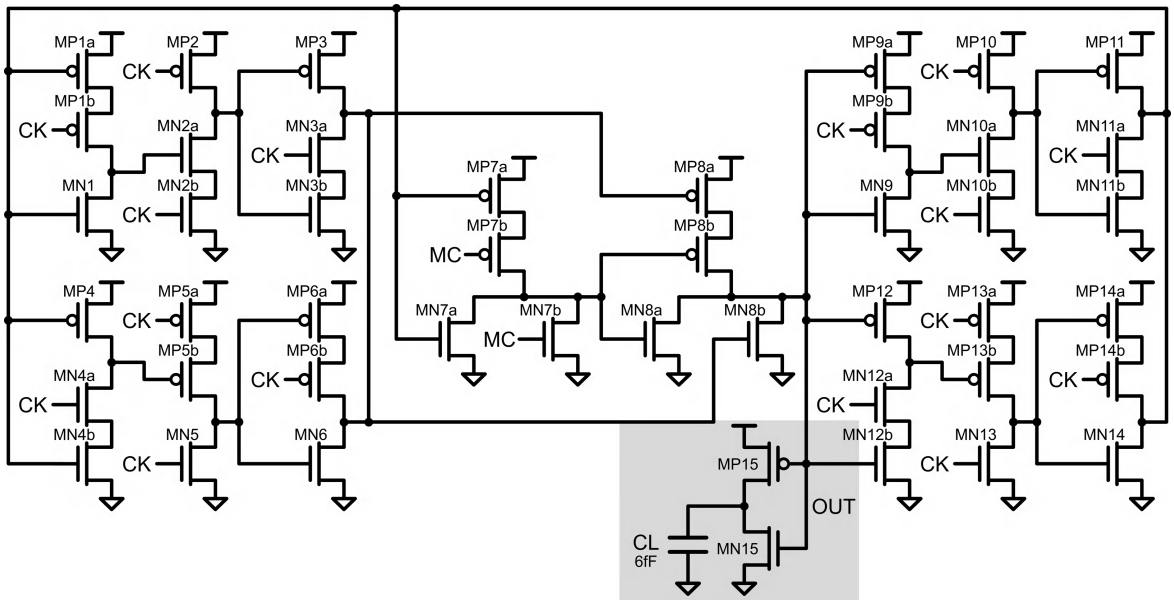
Finite state machines (FSMs) are a cornerstone of digital design and provide a systematic approach to circuit synthesis. Digital counters created using FSM are structured around a set of discrete logic states that are updated at every clock cycle (or every clock transition for DET systems). The state and timing diagrams of the DETFF divide-by-1.5/2 prescaler are shown in Figure 13, where A and B represent the state variables. The counter has two inputs, the clock (CK) and the modulus control signal (MC), and a single output (OUT) that is equal to the state variable B. The MC signal determines the counter modulus, selecting the division ratio 1.5 when LOW, and 2 when HIGH.

Figure 13 – DETFF Divide-by-1.5/2 Prescaler:
(a) state and (b) timing diagrams.



By following standard FSM synthesis method, the DETFF divide-by-1.5/2 prescaler depicted in Figure 14 was derived from the state diagram in Figure 13(a). The circuit output was taken from the input of register B to reduce loading and improve overall speed. The DETFF blocks used in the design are from [31] which proposes two different TSPC DETFFs suitable for high-speed operation. For this prescaler, as well as all subsequent prescaler designs in this work, the circuit load was modeled as a CMOS inverter with a 6 fF capacitor at its output. This generic load block, highlighted with gray shading in all circuit schematics, mimics the input clock capacitance of a TSPC flip-flop.

Figure 14 – DETFF Divide-by-1.5/2 Prescaler Schematic.

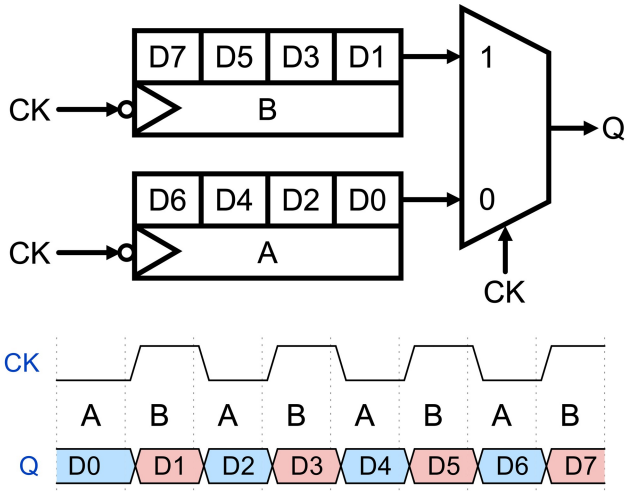


3.3 Original DDR Tecnique

The double data rate (DDR) technique for circuit design was first introduced in [8] and further elaborated in [9]. The original paper presented the idea of alternately selecting the output of two state registers based on the clock phase to form the system output,

allowing for double data throughput without increasing the clock frequency. The DDR technique offers a significant advantage over traditional DETFF designs as only the output merging circuitry is required to operate at twice the clock frequency, resulting in lower power consumption [9]. Figure 15 illustrates a simple example of how DDR can be used to double the data rate of a digital shift register. In this example, register A retains all even data bits and is selected when the clock signal (CK) is LOW, while register B holds all odd data bits and is selected when CK is HIGH.

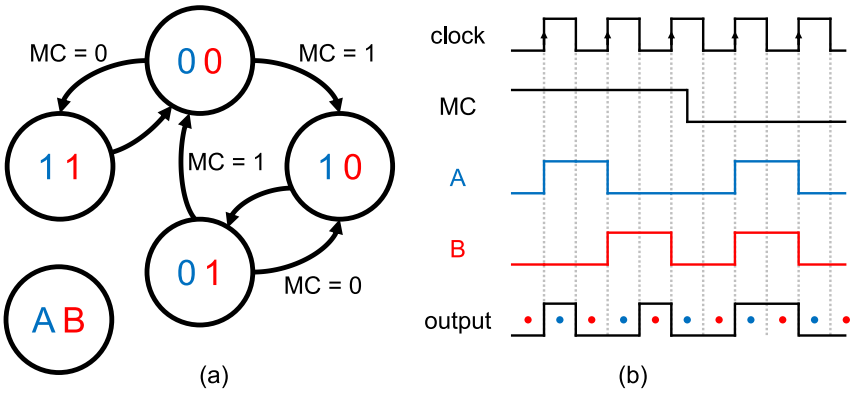
Figure 15 – DDR Shift Register Diagram.



The DDR technique is quite flexible and can be applied to both integer and fractional prescaler designs. The original publication shows an example of a divide-by-4/5 prescaler implemented using DDR where the counter operates at half the clock frequency while the output merging circuitry restores the data rate [9]. To achieve a half-integer division step size, an FSM with an odd-length loop should be designed. This condition is necessary for the system to complete a counting loop in a non-integer number of clock cycles, resulting in a fractional division ratio. For example, a divide-by-1.5 prescaler requires an output sequence that repeats every three clock transitions.

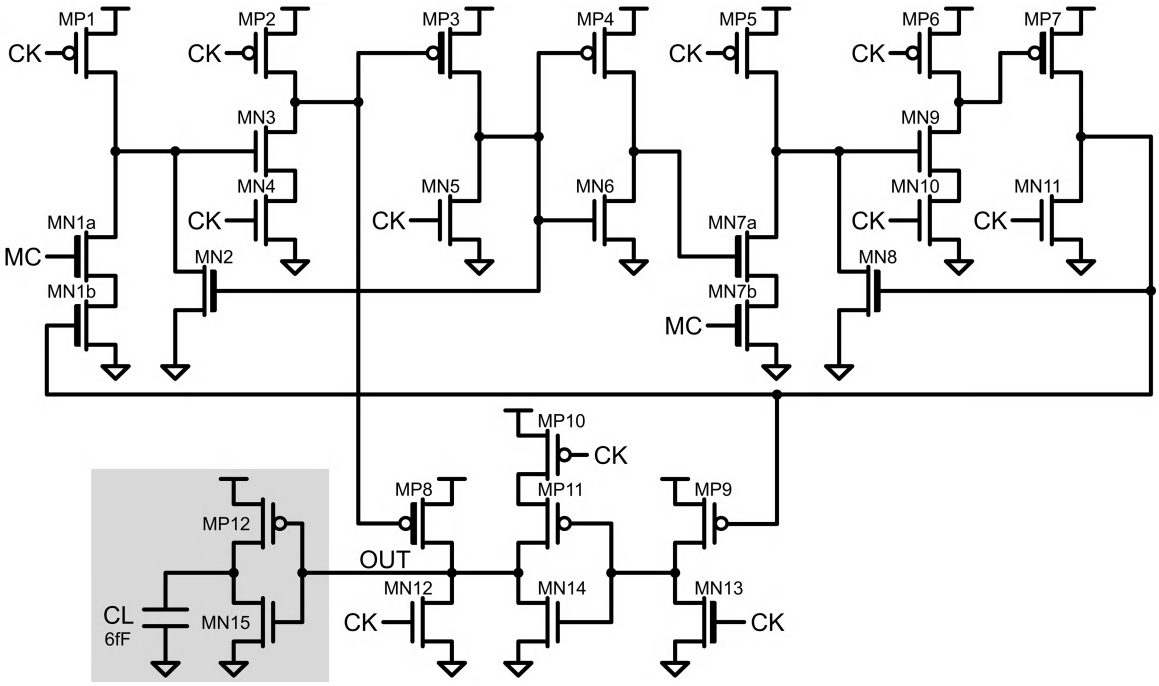
The FSM and timing diagrams of the DDR divide-by-1.5/2 prescaler are depicted in Figure 16. The output signal is formed through the combination of the state variables A and B, following the value in A when the clock signal is HIGH and B otherwise. Blue and red dots were added to the output timing diagram in Figure 16(b) to emphasize the alternating behavior of the system between the state registers A and B, respectively. The presented FSM does not have idle states and is immune to initialization problems common among synchronous systems. Additionally, the FSM control logic allows the Modulus Control (MC) to switch between the division ratios of 1.5 and 2 at any given time, not requiring multiple cycles to change the counter behavior.

Figure 16 – DDR Divide-by-1.5/2 Prescaler:
(a) state and (b) timing diagrams.



From the diagrams presented in Figure 16, the circuit shown in Figure 17 was conceived by following the traditional FSM synthesis method. The logic cells required to build the digital counter were incorporated into the flip-flops to improve speed and reduce transistor count. As in the previous schematics, thick gate MOSFETs represent stronger devices with higher current capability. The flip-flops with embedded logic are depicted at the top of Figure 17, while the DDR merging circuitry is at the bottom with the load block. Both TSPC [22] and extended-TSPC (E-TSPC) [8] structures were used in the design to maximize circuit speed and balance power consumption.

Figure 17 – DDR divide-by-1.5/2 prescaler schematic.



3.4 Balanced DDR Tecnique

A major drawback found while simulating the previous implementations was a non-negligible output period modulation caused by the unbalanced output stages of both the DETFF and DDR methods. Since these circuits rely on NMOS and PMOS data chains* to operate on both clock edges, differences in propagation delay and driving strength accumulate, causing the output frequency to alternate between two values around the expected value. This modulation phenomenon can negatively impact the phase noise performance of the circuit and may counteract the benefits of using a half-integer step frequency divider.

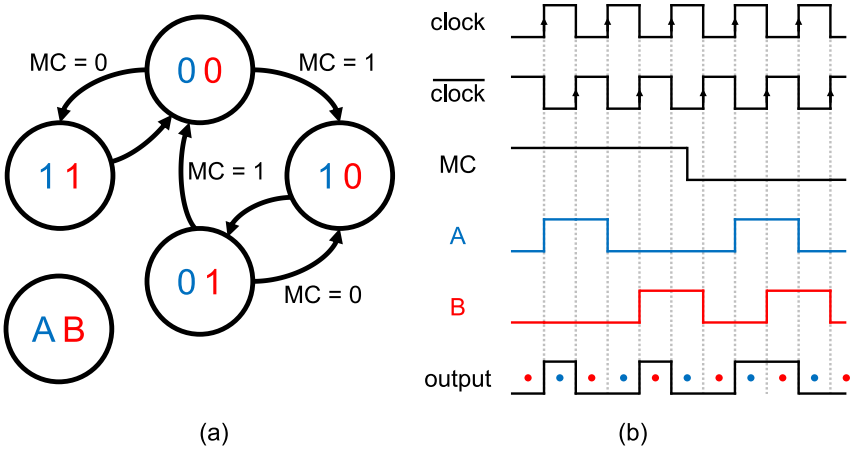
To overcome the limitations of the previous methods, this work proposes a novel balanced design approach that enhances overall circuit performance and reduces transistor count. The balanced DDR (B-DDR) technique is an extension of the original DDR method that utilizes a two-phase clock to drive the prescaler instead of the traditional single-phase clock. This modification enables the use of symmetrical output merging circuitry that greatly reduces the unwanted period modulation effect. Given that voltage-controlled oscillators (VCOs) utilized in high-frequency synthesizers are typically based on differential LC tank or differential ring oscillators, a two-phase clock is expected to be available. Therefore, the B-DDR approach does not introduce further design challenges and offers the additional benefit of facilitating the necessary load balancing of the VCO differential outputs.

The main idea behind B-DDR is to design an FSM where both the original clock (CK) and its complementary (CKB) are used to trigger different storage elements within the circuit. This two-phase clock approach is similar to employing rising and falling edge-triggered flip-flops to design a dual-edge-triggered digital system, but it has the key advantage of not requiring mixing NMOS and PMOS data chains, resulting in symmetrical output circuitry. The more symmetrical the final circuit, the lower the output period modulation, as every data path has similar propagation delay and driving strength. The state and timing diagrams of the B-DDR divide-by-1.5/2 prescaler are shown in Figure 18. As in the previous case, the output is formed by the combination of the state variables A and B, as illustrated by the colored dots in the timing diagram.

The synthesis of B-DDR circuits requires a slightly different method than traditional FSM design. In order to handle two update-triggering signals in the same system, the original state diagram shown in Figure 18 should be divided into two smaller FSMs, each corresponding to one clock phase. In this new configuration, the internal state values stored in one phase should be treated as input signals for the FSM working in the complementary phase. This approach simplifies the original two-phase design into two standard FSM

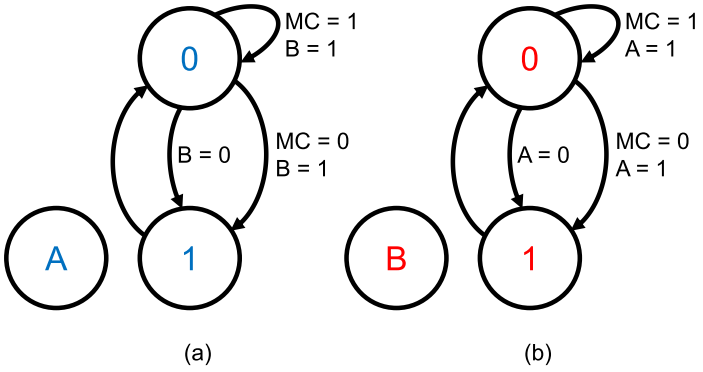
* An NMOS or PMOS data chain is a noncyclical signal propagation path mainly comprised of N-type or P-type structures, respectively [8].

Figure 18 – B-DDR Divide-by-1.5/2 Prescaler:
(a) state and (b) timing diagrams.



syntheses. Figure 19 illustrates this divide-and-conquer principle applied to the B-DDR prescaler design. Since the internal states are updated at different timestamps, selecting a leading signal as the reference is required for correct circuit synthesis. In Figure 19, variable A was chosen as the leading variable, so the variable B state diagram was drawn considering A one clock cycle ahead of B in the Figure 18(a) original state diagram.

Figure 19 – B-DDR Divide-by-1.5/2 Prescaler:
Split State Diagrams.

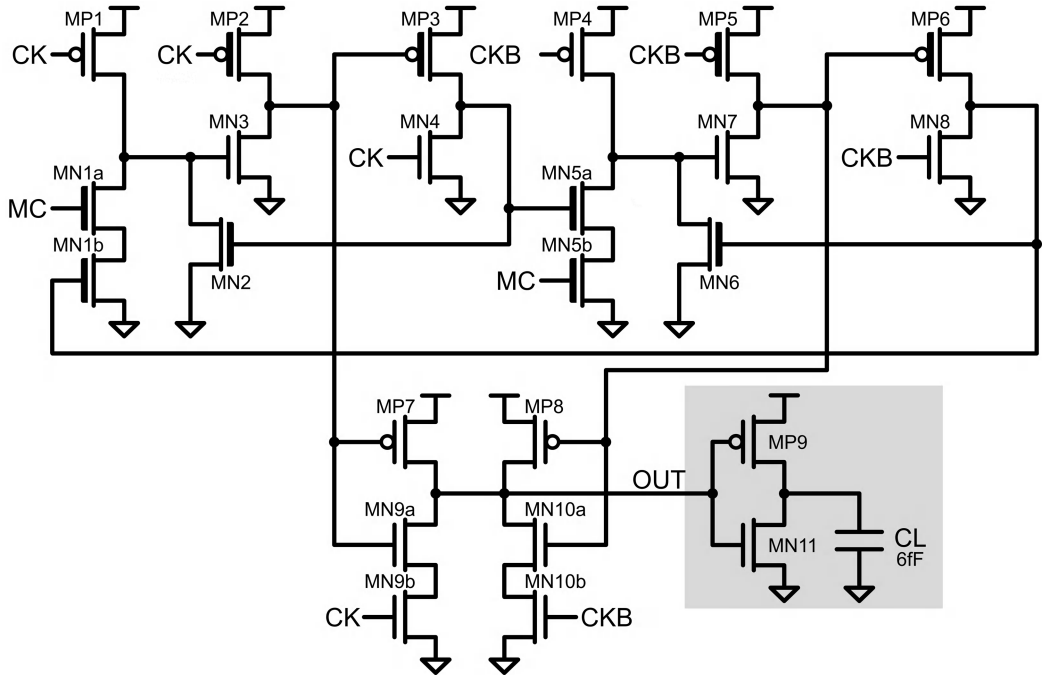


The split diagrams in Figure 19 may seem counterintuitive at first glance, but they provide a powerful tool for simplifying FSM synthesis. The procedure to go from the state diagram in Figure 18(a) to the split diagram of A in Figure 19 is quite straightforward and similar to conventional FSM, with the only caveat of considering B as an external input rather than an internal state. The same principle does not apply to the state diagram of B, as it was previously defined as the lagging variable of the system. In this case, some abstraction is required when analyzing the diagram in Figure 18(a) to account for the intermediary (undrawn) states. For example, if the system starts in state 00 and variable A transitions to 1 on the first rising clock edge, the transition of B should be considered with respect to state 10 rather than 00, since A is the leading variable and has already

transitioned, whereas B lags and will transition only on the following falling clock edge. This same approach should be applied for all possible B states while considering A an external input alongside the MC signal.

Based on the split diagrams presented in Figure 19, the circuits depicted in Figure 20 and Figure 21 were designed using the conventional FSM synthesis method. These circuits are nearly identical, but Figure 20 (B-DDRv1) employs four NMOS-like blocks, while Figure 21 (B-DDRv2) utilizes only two. This subtle modification makes the B-DDRv1 circuit more suitable for high-speed applications, as the ratioed blocks offer faster response times at the expense of higher power consumption. On the other hand, the B-DDRv2 version is a lower-power alternative with a lower maximum speed. In both implementations, the output merging circuitry is symmetrical and minimizes the period modulation. As in the previous schematics, thick gate MOSFETs represent stronger devices with higher current capability.

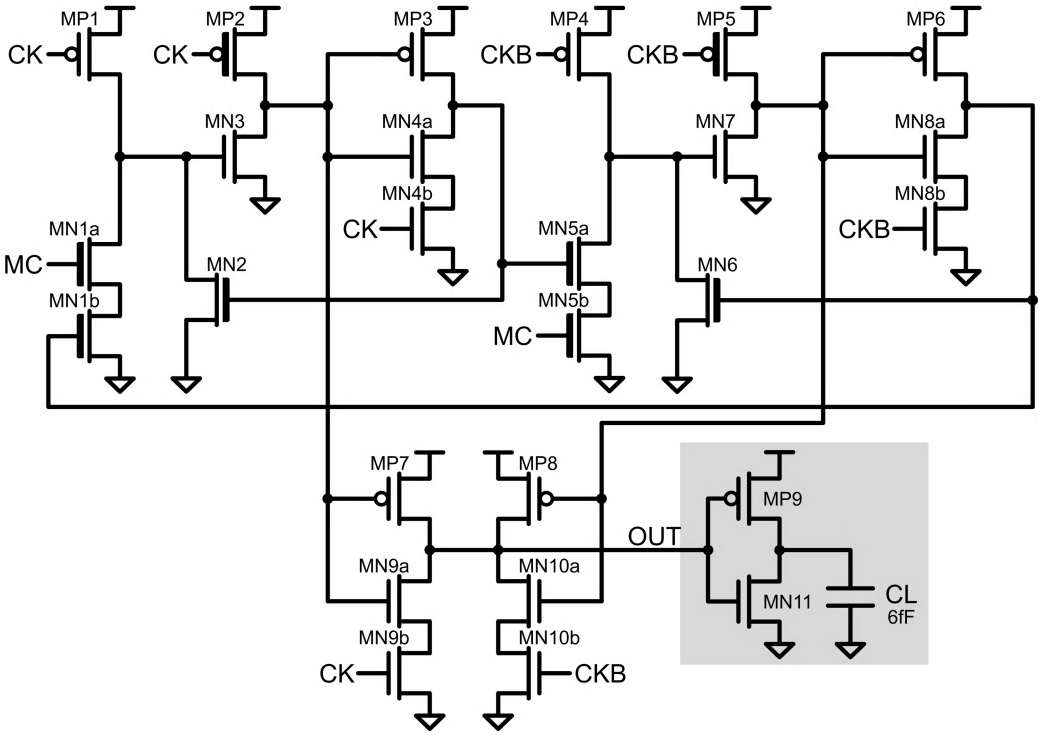
Figure 20 – B-DDRv1 Divide-by-1.5/2 Prescaler Schematic.



3.5 Transistor Sizing and Circuit Optimization

All five prescaler circuits presented so far lack information regarding transistor sizing and only provide a rough estimate of device strength, indicated by the symbolic usage of thick-gate MOSFETs in the schematics. This lack of detailed information may seem unhelpful, but knowing the relative device strength sets general boundary conditions for transistor sizing that allow for customized circuit design. As discussed in previous sections, the performance of both TSPC and E-TSPC structures heavily depends on

Figure 21 – B-DDRv2 Divide-by-1.5/2 Prescaler Schematic.



device sizing, and a circuit designed for a specific operating frequency may not function properly outside its range. Therefore, manually sizing the MOSFETs is unsuitable for a fair comparison among the proposed designs, and a more robust and unbiased approach is required to attain fair results.

A transistor sizing and circuit optimization framework named CirOp was selected to ensure a fair design of all analyzed circuits. The framework operation is based on device sizing through metaheuristics, where each design is simulated and its performance rated [32]. Several metaheuristics algorithms are available in the framework, and a combination of particle swarm optimization (PSO) and simulated annealing (SA) was selected to optimize the prescaler circuits [33]. For each different topology, the search space was defined by parameterizing the widths and, in some cases, the lengths of the MOSFETs in the circuit and allowing these dimensions to vary within a specified range. The design score was calculated based on the following attributes: the maximum operating frequency, power consumption, gate area of the clocked transistors, total transistors area, period modulation, and output duty cycle. All circuits were implemented in the Taiwan Semiconductor Manufacturing Company (TSMC) 65 nm CMOS process and compared through HSPICE simulations using a typical BSIM4 MOSFET model.

The circuit simulations were conducted using a 1.2 V power supply and a square wave clock signal. The clock signal was configured for 15% rise and fall times relative to its period. All designs operated under identical load conditions, which consisted of a CMOS inverter with a total combined transistor width of 2.0 µm and a capacitor of 6.0 fF

placed at its output, as illustrated in the schematics. A design was considered valid if it achieved a minimum working frequency of 2.0 GHz and maintained a minimal output voltage swing ranging from 10% to 90% of the supply voltage across the capacitor load. Additional information regarding the optimization variables can be found in Appendix A.

The key benefit of using a framework for transistor sizing and circuit optimization is that the same computational effort is applied to the design of every circuit. This single aspect makes the comparison of different circuits subjected to the same framework conditions much fairer and unbiased, as no human decisions are present in the loop. For the prescaler circuits, the optimization started by first running the designs through a PSO algorithm to find possible candidates for the best performance. Next, a modified simulated annealing algorithm was used to further refine the PSO results [33, 34]. The stopping criteria for both optimization stages were set to four thousand different analyzed circuits.

3.5.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an optimization algorithm inspired by the collective intelligence observed in natural systems, such as bird flocks and fish shoals. The algorithm was first introduced by Kennedy and Eberhart in 1995 [34] as a computational method for finding optimal regions within the search space through the interaction of individuals in a population. These individual agents, referred to as particles, evaluate a cost function locally and share the results throughout the swarm, influencing the overall group behavior. The interactions among particles lead to the emergence of a complex global intelligence that allows the swarm to solve optimization problems effectively.

Each particle in PSO is initialized with a random position and velocity, and it evaluates a cost function using its coordinates as input values. As the agents start to randomly explore the solution space, they broadcast their findings to the entire population. If a particle discovers a more promising path that may lead to a better solution, it prompts its peers to adjust their trajectories toward this new direction [35]. This process is analogous to a flock of birds searching for food. As one bird finds a food source, it signals to the others, leading the entire flock towards the source. Through this mechanism, PSO seeks the optimal solution for a complex problem by navigating the solution space. The collective effort of all particles allows the swarm to converge on the best possible solution efficiently, exploring a significant part of all possible solutions in the search process [34].

For applications of the PSO algorithm in electronic design optimization, circuit parameters are encoded as particle coordinates in an n-dimensional space. Instead of evaluating a mathematical cost function based on these n-coordinates, the PSO algorithm employs SPICE simulations to assess circuit behavior for each combination of design parameters represented by the particle positions. With the solution space encompassing a wide range of potential circuit configurations, the simulation results guide the particles

toward the ideal circuit design over successive iterations. Although the performance metrics and parameterized variables may differ between applications, the general approach involves strategically simulating the design across a broad range of parameter combinations to achieve near-optimal results [33].

3.5.2 Simulated Annealing

Simulated Annealing (SA) is a local search optimization algorithm proposed by Kirkpatrick, Gelatt, and Vecchi in 1983 [36] inspired by the annealing process in metallurgy. Annealing is a heat treatment commonly used in materials science to alter the physical and chemical properties of crystalline solids. The annealing process involves heating a material to a very high temperature and then slowly cooling it down, allowing its atoms to reach a state of low energy and minimal internal stress. At the beginning, the atoms move freely and randomly due to the high-energy state of the system. As the temperature decreases, their movements become less random and more regular, eventually stabilizing around the final lattice structure [33]. This principle of allowing erratic movements of atoms at first, and then gradually reducing their freedom as the temperature decreases, is mimicked in the SA algorithm.

The SA algorithm is particularly effective for solving problems with complex solution spaces that contain many local optima. The algorithm starts with a high temperature that allows for the generation of fairly random new solutions. As the temperature decreases, the algorithm becomes more selective, reducing the likelihood of accepting significantly worse solutions and focusing on converging towards a near-optimal state. Although counterintuitive, accepting temporarily worse solutions is key for an effective local search, as it helps prevent the algorithm from getting trapped in suboptimal regions of the solution space. New possible solutions are generated by perturbing the current solution, and thus introducing some randomness ensures a broader exploration during the search. The final solution is approached as the temperature gets very low, at which point the algorithm primarily selects improvements or minor deteriorations, thereby refining the solution to a near-optimal value [33, 36].

For applying the SA algorithm in electronic design optimization, a set of circuit parameters is interpreted as a given state of atoms within the simulated crystalline structure, and the ‘energy level’ of this circuit configuration is calculated through SPICE simulations. Although unconventional, the circuit performance is inversely graded based on the simulation results, as the ideal circuit configuration should achieve the lowest possible value to maintain some consistency with the real annealing process. The SA temperature variable, which influences the likelihood of the algorithm accepting suboptimal solutions, is typically related to the total number of optimization iterations, decreasing as more circuit configurations are evaluated [33].

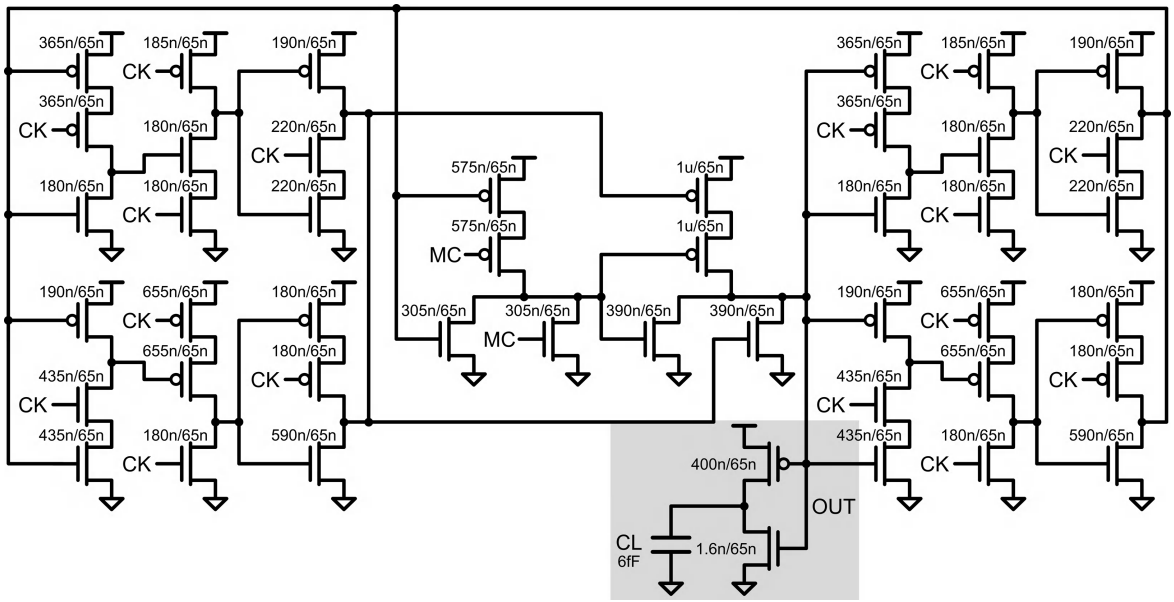
4 RESULTS AND DISCUSSION

The following chapter presents the main results obtained throughout this research, along with a comprehensive analysis and discussion of these findings. The prescaler topologies previously introduced were optimized using the metaheuristics framework, and their performance was evaluated through SPICE simulations. All four circuits were designed to reach 4.0 GHz clock operation, the maximum speed supported by the TSPC DETFF design. Additionally, they were optimized to minimize power consumption, maintain the total gate area of the clocked transistors lower than 0.4 µm², exhibit minimal period modulation, and ensure a correct duty cycle. Key performance metrics, such as maximum operating frequency, power consumption, period modulation, transistor count, and the total clocked area are highlighted for each topology.

4.1 DETFF Divide-by-1.5/2 Prescaler

The DETFF divide-by-1.5/2 prescaler circuit is the most standard implementation of a DET-based fractional divider and was the first topology to be optimized using the metaheuristics framework. Figure 22 redraws the previously presented schematic with the optimal transistor sizing for operation at a 4.0 GHz clock, which is the maximum supported by this design in a 65 nm CMOS process. Multiple attempts were made to achieve higher operating frequencies, but the optimization constraints limited the design to a maximum 4.0 GHz clock. Therefore, this frequency was selected as the benchmark for subsequent designs to ensure a fair comparison across different topologies.

Figure 22 – DETFF Prescaler Optimized for 4.0 GHz.
(transistor sizes W/L).



The final circuit operates within a frequency range of 1.0 GHz to 4.0 GHz, consuming an average power of 80.6 µW at a 4.0 GHz clock under a 1.2 V supply. Figure 23 illustrates the operation of the sized circuit at 4.0 GHz in both the divide-by-1.5 (div15) and divide-by-2 (div2) modes. Figure 24 shows the variation of the output period over time for the prescaler in divide-by-1.5 mode, with T_r representing the period measured from the signal rising edge and T_f from the falling edge. The graph clearly demonstrates the period modulation phenomenon, where the output period alternates between two values around the expected period. The presented data indicates a 5.7 ps worst-case period deviation while in steady-state operation, representing a 1.52% error with respect to the ideal 375 ps output period. A period modulation inferior to 1 ps was observed in the system while operating in the divide-by-2 mode.

Figure 23 – DETFF Prescaler Operation at 4.0 GHz.

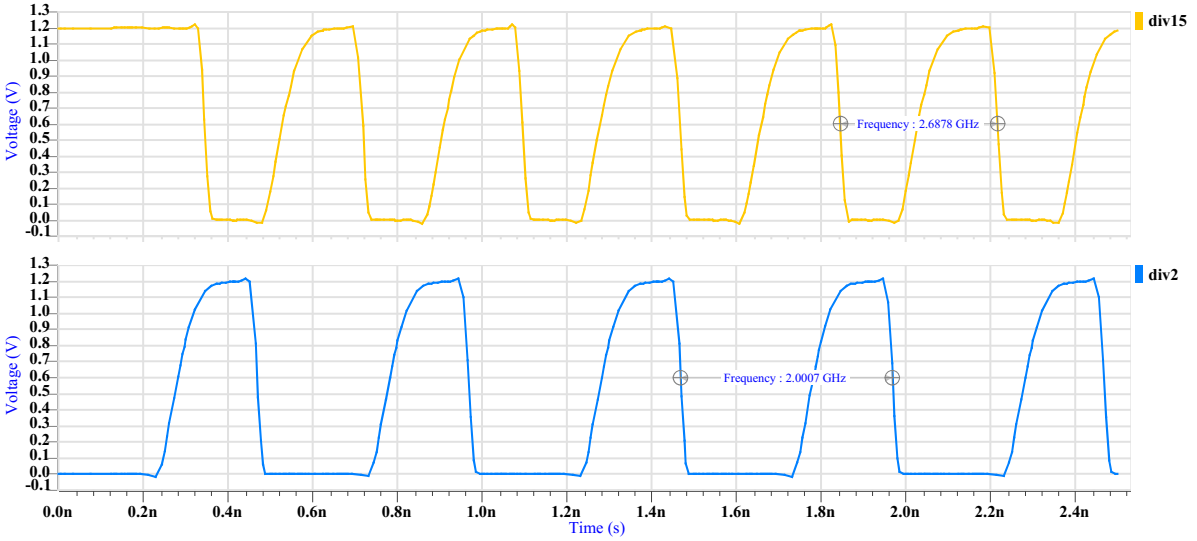
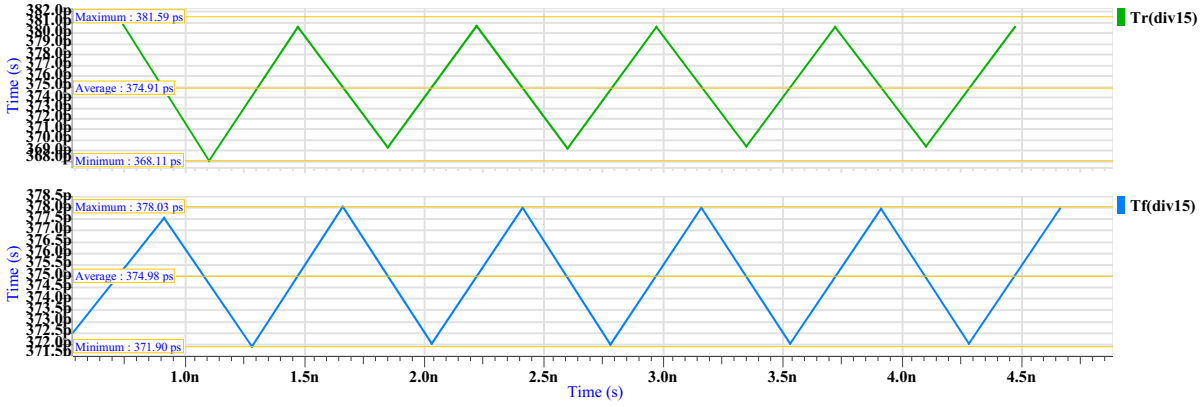


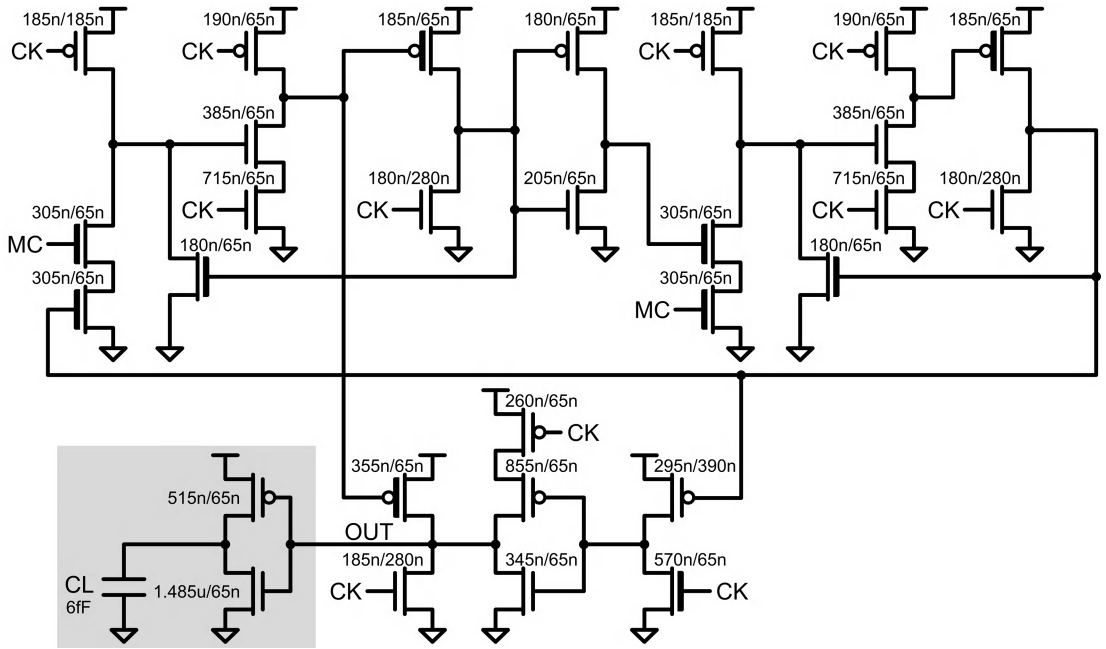
Figure 24 – DETFF Prescaler Period Modulation.



4.2 DDR Divide-by-1.5/2 Prescaler

The DDR divide-by-1.5/2 prescaler circuit schematic with the optimal transistor sizing for 4.0 GHz clock operation is shown in Figure 25. Both the width and length of the lower transconductance transistors in the E-TSPC ratioed structures were sized through the metaheuristics framework to enhance performance, as increasing the MOSFET length decreases the overall device strength. Although the main optimization was constrained to a 4.0 GHz frequency, some additional runs were performed to estimate the maximum speed supported by this design. Preliminary data suggests that the DDR prescaler could easily achieve 13.5 GHz operation and possibly more if certain optimization constraints, such as the total clocked gate area, were relaxed.

Figure 25 – DDR Prescaler Optimized for 4.0 GHz.
(transistor sizes W/L).



The circuit depicted in Figure 26 operates effectively across a frequency range of 1.0 GHz to 6.0 GHz, with an average power consumption of 90.0 µW at 4.0 GHz under a 1.2 V supply. Figure 27 illustrates the output waveform of the prescaler in both the divide-by-1.5 (div15) and divide-by-2 (div2) modes while operating with a 4.0 GHz input clock. Figure 28 displays the output period variation over time for the circuit in divide-by-1.5 mode, where T_r is the period measured with respect to the signal rising edge and T_f from the falling edge. From this graph, the worst period deviation in steady-state is 10.7 ps, corresponding to a 2.85% error relative to the ideal 375 ps output. Negligible period modulation was measured when the circuit was set in divide-by-2 mode.

Figure 26 – DDR Prescaler Operation at 4.0 GHz.

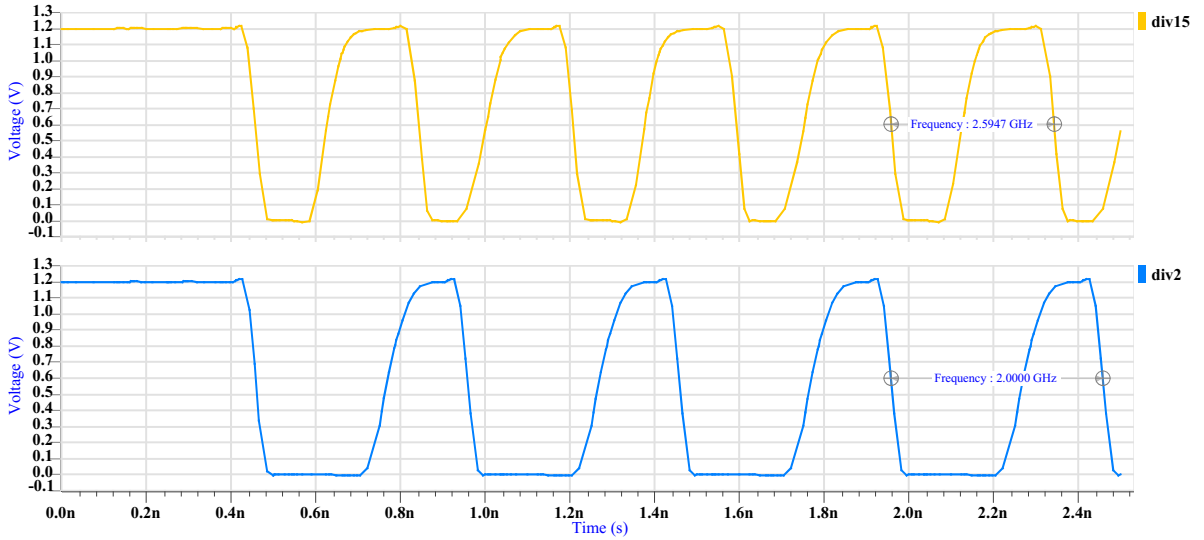
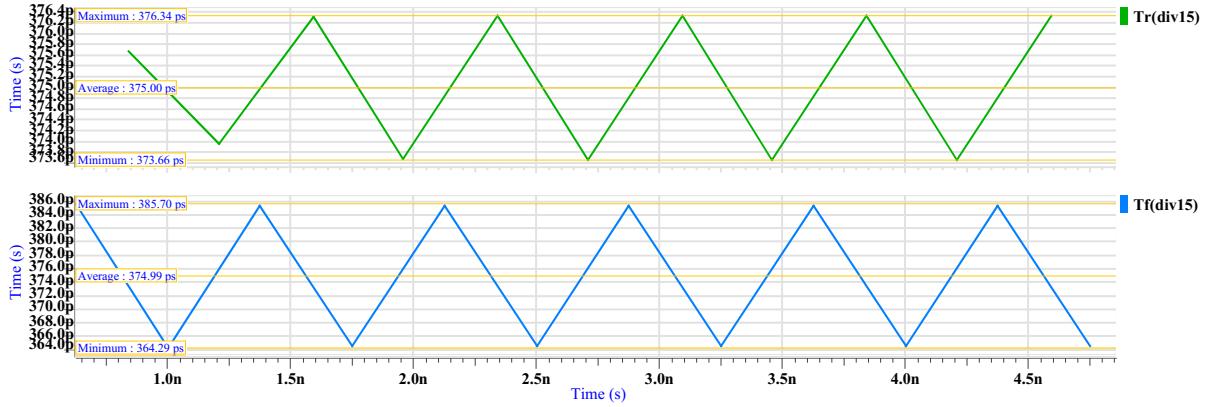


Figure 27 – DDR Prescaler Period Modulation.

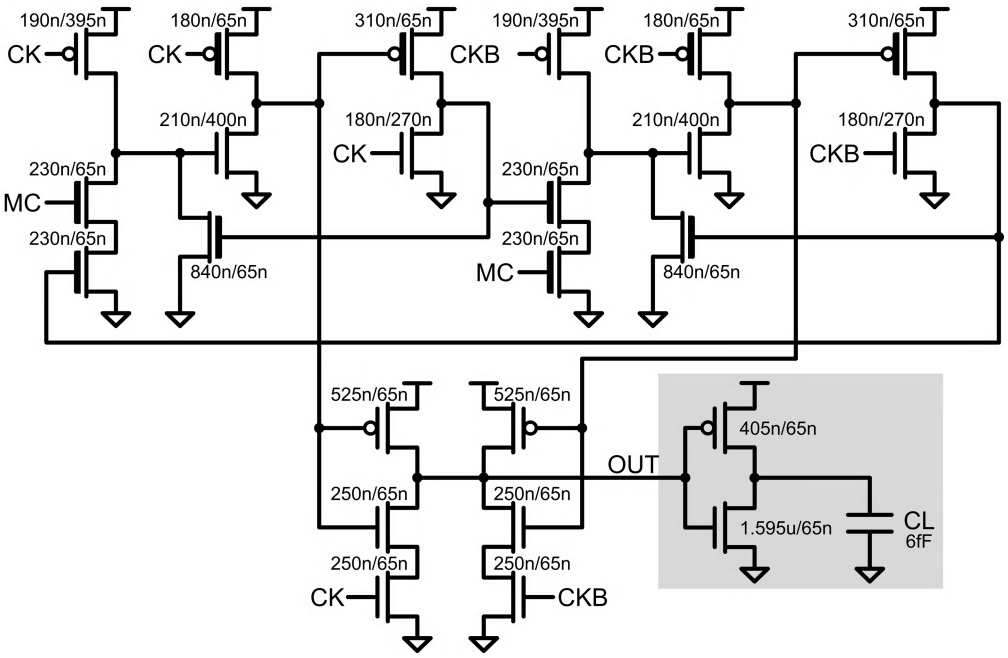


4.3 B-DDRv1 Divide-by-1.5/2 Prescaler

Figure 28 illustrates the redrawn B-DDRv1 divide-by-1.5/2 prescaler circuit with the optimal transistor sizing for a 4.0 GHz clock operation. Similar to the previous topology, the widths and lengths of the MOSFETs in the ratioed E-TSPC structures were sized using the metaheuristics framework to achieve better energy efficiency with a lower area. Further investigations were carried out to determine the maximum operating frequency of the topology by re-running optimizations with a focus on achieving higher circuit speed. Preliminary data suggests that the B-DDRv1 design can handle an input frequency of up to 15.0 GHz and potentially reach even higher frequencies with additional sizing iterations.

The B-DDRv1 circuit operates within the frequency range of 2.0 GHz to 4.0 GHz, with an average power consumption of 62.3 µW at an input clock frequency of 4.0 GHz and a 1.2 V supply voltage. Figure 29 shows the output waveform in both the divide-by-1.5 (div15) and divide-by-2 (div2) modes. Figure 30 presents the circuit output period variation over time for the prescaler configured to 1.5 division ratio. The T_r waveform represents

Figure 28 – B-DDRv1 Prescaler Optimized for 4.0 GHz.
(transistor sizes W/L).



the output period measured between two successive rising edges, while the T_f is between two consecutive falling edges. As expected, the B-DDR symmetry significantly reduces the steady-state period modulation effect, resulting in a maximum deviation of 0.2 ps from the ideal period of 375 ps, which translates to a period error of 0.05%. The transient deviation visible in Figure 30 was not taken into account for any of the analyzed circuits, as there is no straightforward way to determine if its occurrence is due to a faulty circuit characteristic or to simulation initial conditions. A negligible period modulation of less than 0.1 ps was observed when the circuit operated in the divide-by-2 mode.

Figure 29 – B-DDRv1 Prescaler Operation at 4.0 GHz.

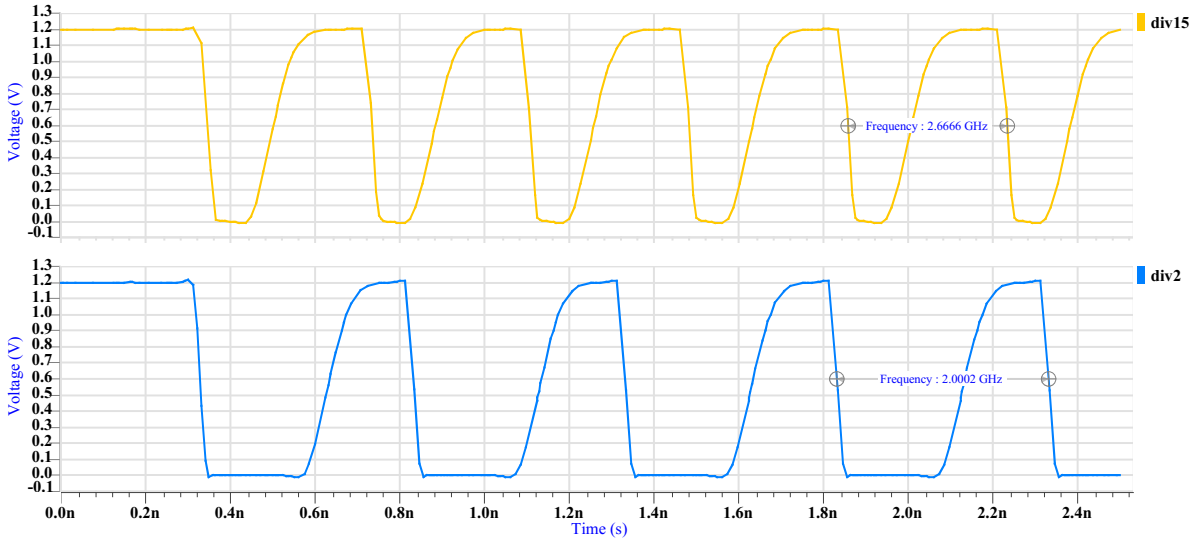
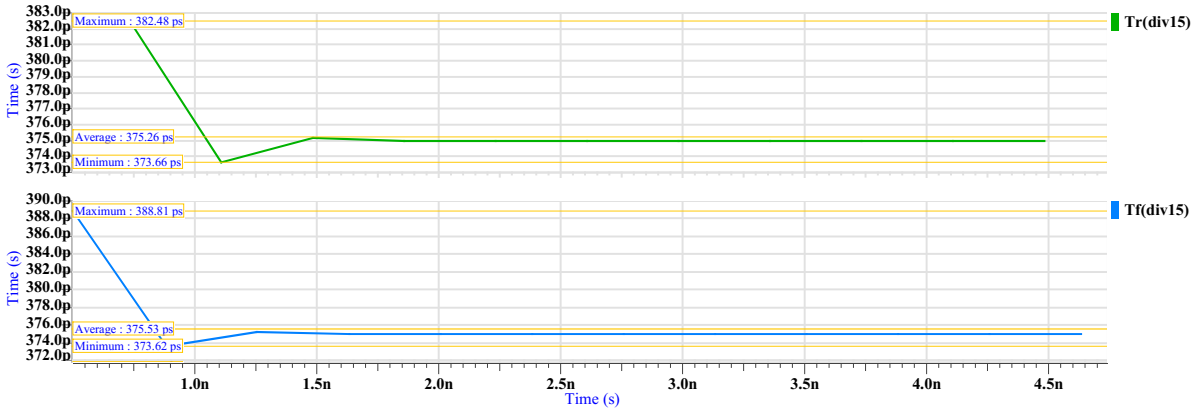


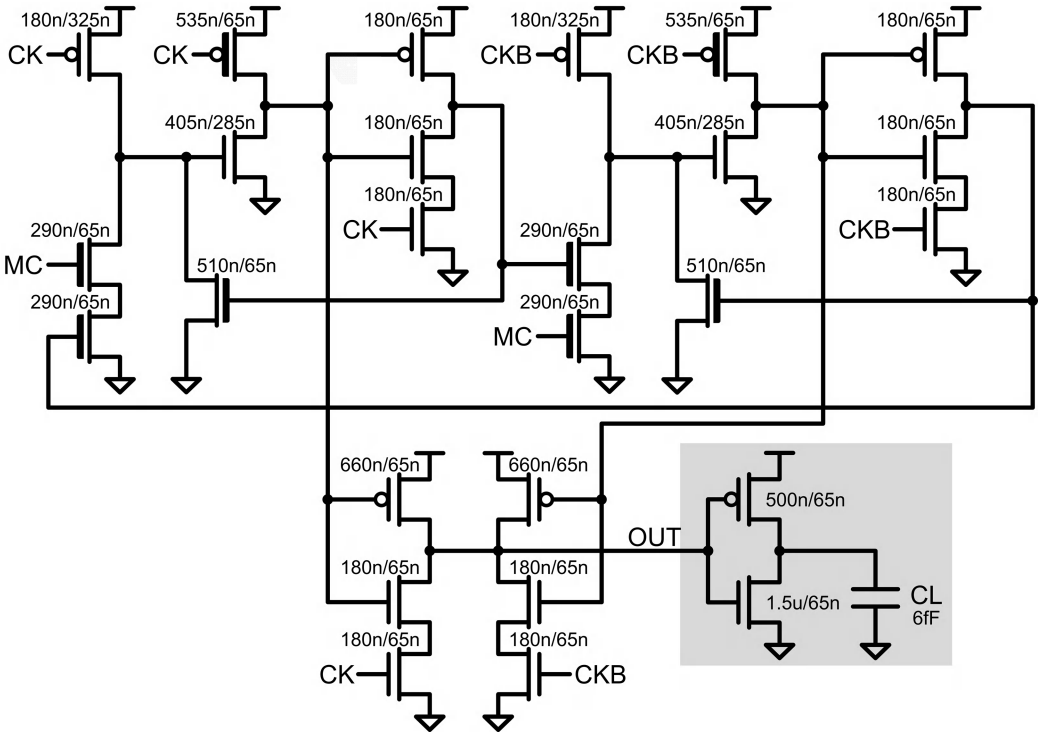
Figure 30 – B-DDRv1 Prescaler Period Modulation.



4.4 B-DDRv2 Divide-by-1.5/2 Prescaler

The redrawn B-DDRv2 divide-by-1.5/2 prescaler schematic with optimal transistor sizing for 4.0 GHz clock operation is presented in Figure 31. For the ratioed E-TSPC structures, both the MOSFET width and length parameters were optimized using the metaheuristics framework, as was done in previous designs. Beyond the main optimization at 4.0 GHz, additional runs were conducted to check the maximum operating speed of the topology. According to preliminary results, the B-DDRv2 seems to achieve a staggering 16.5 GHz clock frequency operation, which may or may not be the highest attainable performance for this design in a 65 nm CMOS technology.

Figure 31 – B-DDRv2 Prescaler Optimized for 4.0 GHz.
(transistor sizes W/L).



The B-DDRv2 circuit operates effectively within an input frequency range of 1.0 GHz to 5.0 GHz, consuming an average power of 54.2 µW at a 4.0 GHz frequency under a 1.2 V power supply. Figure 32 presents the output waveform for the circuit operating at a 4.0 GHz clock frequency in both the divide-by-1.5 (div15) and divide-by-2 (div2) modes. Figure 33 depicts the output period variation over time for the prescaler in the divide-by-2 mode. Here, T_r denotes the period measured from rising edges, while T_f represents the period measured from falling edges. Unlike previous designs, the B-DDRv2 exhibited a greater output period modulation in the divide-by-2 mode, with a maximum deviation of 2.1 ps. This value corresponds to a 0.42% error relative to the ideal 500 ps period. The output period behavior of the B-DDRv2 at a division ratio of 1.5 was similar to that of the B-DDRv1, showing a maximum steady-state period deviation of 0.2 ps (0.05% error).

Figure 32 – B-DDRv2 Prescaler Operation at 4.0 GHz.

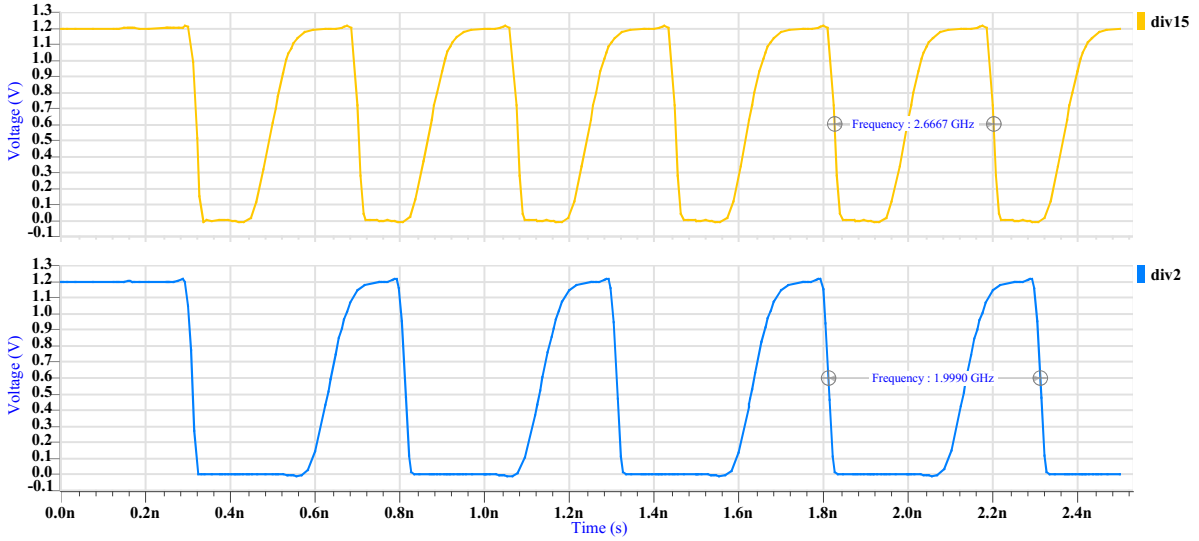
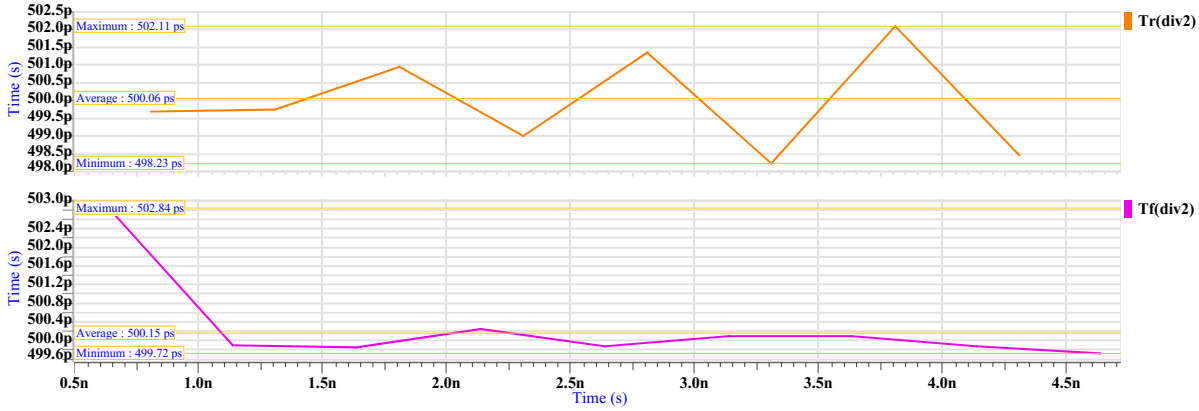


Figure 33 – B-DDRv2 Prescaler Period Modulation.



4.5 Summary and Comparison

Figure 34 presents the average power consumption versus operating frequency for the four prescaler circuits optimized to work at a 4.0 GHz clock frequency. The average power consumption was calculated by taking the mean power consumption of the circuits operating in divide-by-1.5 and divide-by-2 modes. Only data within the working frequency range of each design was considered valid for this analysis. Other relevant numerical data, including the number of optimized variables, the transistor count, the total transistors area, the total clocked area, the worst period modulation at 4.0 GHz, and the maximum speed (which was obtained through additional optimizations, resulting in new circuits), are presented in Table 2. Preliminary high-speed operation data is available in Appendix B.

Figure 34 – Power Consumption versus Frequency.
(designs optimized for 4.0 GHz)

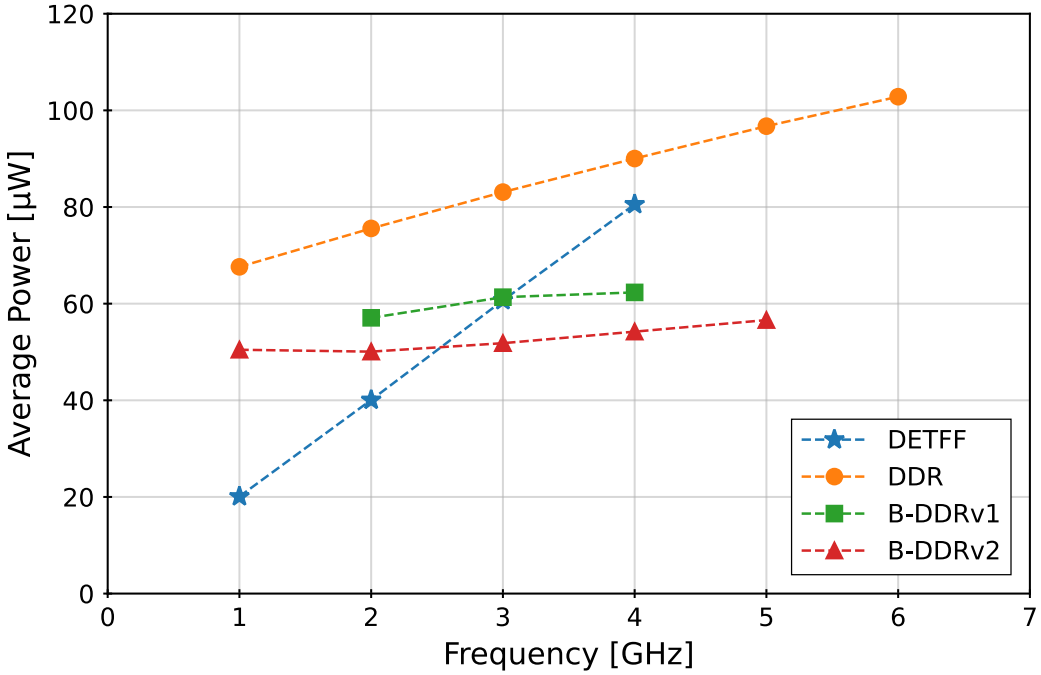


Table 2 – Relevant Numerical Results.

Design	Number of optimized variables	Transistor count	Total Transistors Area (µm²)	Clocked Area (µm²)	Worst PM at 4.0 GHz	Maximum speed (GHz)
DETFF	17	44	4.16	0.31	1.52%	4.0
DDR	21	27	2.51	0.39	2.85%	13.5**
B-DDRv1	13	22	2.14	0.30*	0.05%	15.0**
B-DDRv2	12	24	2.27	0.23*	0.42%	16.5**

* Total area considering CK + CKB.

** Preliminary data.

From the simulation results presented in Figure 34 and Table 2, and taking the classical DETFF divide-by-1.5/2 prescaler as a benchmark, the following key considerations can be drawn:

(1) The DETFF prescaler exhibits the poorest performance metrics among the designs reviewed when considering transistor count, total transistors area, and maximum operating speed. While it offers a very simple design approach, it lacks the optimizations required for high-performance applications. Based on the available data, the DETFF prescaler appears to be the most energy-efficient in low-speed applications. However, since all designs were optimized for 4.0 GHz operation, the other topologies may exhibit better power consumption metrics if sized for lower operating frequencies.

(2) The DDR prescaler shows the highest power consumption and the worst period modulation at 4.0 GHz across all evaluated topologies. Its power consumption is 11.8% higher than the DETFF circuit at 4.0 GHz, while its maximum period deviation is almost twice that of the DETFF circuit. Despite this, the DDR design offers two major improvements: a 39% reduction in total transistors area and a 237.5% increase in maximum operating frequency. Additionally, the DDR design was the only one to achieve an operating frequency of 6.0 GHz, demonstrating good circuit flexibility even though it was optimized for a 4.0 GHz clock. Overall, the DDR technique may not yield the most power-efficient designs at lower frequencies, but it offers the best option for fractional frequency division with single-phase clocking above the 4.0 GHz mark. For low-speed applications requiring good energy efficiency, a possible improvement would be to partially or completely replace the ratioed E-TSPC structures in the design with TSPC-based ones to minimize static power consumption.

(3) The prescalers based on the B-DDR technique demonstrated the best results by a significant margin. Although B-DDR is the most complex design method among the compared approaches, the B-DDR circuits exhibited the lowest power consumption at 4.0 GHz, the smallest transistors area, and the highest maximum operating frequencies. The B-DDRv2 version showed a significant reduction of 32.7% in power consumption, a 45% smaller total transistors area, and a gain of 312.5% in the maximum operating frequency when compared to the DETFF implementation. Additionally, both B-DDR versions exhibited small period modulation during steady-state operation. Another noteworthy advantage of the B-DDR approach is its smaller total clocked area, which results in a lower capacitance load for the local oscillator. Considering the two complementary clock phases separately, the capacitance seen by the CK source when connected to the B-DDRv2 prescaler is 63% lower compared to the DETFF circuit. Even accounting for the summed total clocked area, the B-DDRv2 design offers a 26% reduction in total capacitance. Overall, the B-DDR topology appears to be the most suitable for high-performance applications, provided that two complementary clock phases are available in the system.

5 CONCLUSION

This work has presented three new divide-by-1.5/2 prescaler topologies based on the DDR technique and benchmarked their performance against a traditional DETFF implementation. All four prescaler circuits were implemented in the TSMC 65 nm CMOS process and designed with an automated transistor sizing and circuit optimization framework based on metaheuristic algorithms. Particle swarm optimization and simulated annealing were employed during the optimization phase to minimize power consumption and area for a 4.0 GHz target operating frequency. Consistent optimization constraints were applied to all topologies to ensure an unbiased comparison. Simulation results demonstrated that the DDR and B-DDR techniques enabled smaller and faster designs, achieving a 39% to 45% transistors area reduction and delivering speed improvements ranging from 3.37 to 4.12 times the maximum DETFF speed.

The novel B-DDR technique, introduced as an extension of the original DDR method, was employed to effectively minimize the period modulation phenomenon observed during simulations. Numerical data demonstrated that the proposed B-DDR method significantly reduces period variation and facilitates low-power design. Circuit comparisons at 4.0 GHz revealed a substantial power reduction of 22% for B-DDRv1 and 32% for B-DDRv2 designs. The B-DDR technique demonstrated promising results and potential for use in high-performance digital applications, with the limitation of requiring two complementary clock phases in the system.

As suggestions for future work, the following research ideas are proposed: verify the maximum operating speed of the proposed circuits through various metaheuristic optimizations; analyze the advantages and disadvantages of employing more or fewer ratioed blocks in the topologies; draw the prescaler layouts to refine the simulation results and comparisons; verify the behavior of the new prescalers across various temperatures and technology process corners; and investigate the feasibility of implementing other high-performance digital circuits using the B-DDR technique. These research topics represent a small sample of the various questions that emerged during the execution of this work.

REFERENCES

1 RAZAVI, B. **RF Microelectronics (2nd Edition) (Prentice Hall Communications Engineering and Emerging Technologies Series)**. 2nd. ed. USA: Prentice Hall Press, 2011. ISBN 0137134738.

2 LACAITA, A. L.; LEVANTINO, S.; SAMORI, C. **Integrated Frequency Synthesizers for Wireless Systems**. 1st. ed. USA: Cambridge University Press, 2007. ISBN 0521863155.

3 JIN, J. *et al.* Quantization noise suppression in fractional-n plls utilizing glitch-free phase switching multi-modulus frequency divider. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 59, n. 5, p. 926–937, 2012.

4 YANG, Y.-C. *et al.* A quantization noise suppression technique for$deltasigma$fractional-$n$frequency synthesizers. **IEEE Journal of Solid-State Circuits**, v. 41, n. 11, p. 2500–2511, 2006.

5 RAZAVI, B. **RF Microelectronics**. 1st. ed. USA: Prentice-Hall, Inc., 1998. ISBN 0138875715.

6 RILEY, T.; COPELAND, M.; KWASNIEWSKI, T. Delta-sigma modulation in fractional-n frequency synthesis. **IEEE Journal of Solid-State Circuits**, v. 28, n. 5, p. 553–559, 1993.

7 YANG, Y.-C. *et al.* A dual-mode truly modular programmable fractional divider based on a 1/1.5 divider cell. **IEEE Microwave and Wireless Components Letters**, v. 15, n. 11, p. 754–756, 2005.

8 SOARES, J. N.; NOIJE, W. V. A 1.6-ghz dual modulus prescaler using the extended true-single-phase-clock cmos circuit technique (e-tspc). **IEEE Journal of Solid-State Circuits**, v. 34, n. 1, p. 97–102, 1999.

9 NAVARRO, S. J.; NOIJE, W. V. Extended tspc structures with double input/output data throughput for gigahertz cmos circuit design. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 10, n. 3, p. 301–308, 2002.

10 CRANINCKX, J.; STEYAERT, M. A 1.75-ghz/3-v dual-modulus divide-by-128/129 prescaler in 0.7-/spl mu/m cmos. **IEEE Journal of Solid-State Circuits**, v. 31, n. 7, p. 890–897, 1996.

11 WANG, S. *et al.* Power efficient multimodulus programmable frequency divider with half-integer division ratio step size. *In*: **2009 16th IEEE International Conference on Electronics, Circuits and Systems - (ICECS 2009)**. [*S.l.: s.n.*], 2009. p. 739–742.

12 THIRUNARAYANAN, R.; RUFFIEUX, D.; ENZ, C. An injection-locking based programmable fractional frequency divider with 0.2 division step for quantization noise reduction. *In*: **2013 Proceedings of the ESSCIRC (ESSCIRC)**. [*S.l.: s.n.*], 2013. p. 233–236.

13 HU, A. *et al.* A 0.045- to 2.5-ghz frequency synthesizer with tdc-based afc and phase switching multi-modulus divider. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 67, n. 12, p. 4470–4483, 2020.

14 RAZAVI, B. The design of a phase interpolator [the analog mind]. **IEEE Solid-State Circuits Magazine**, v. 15, n. 4, p. 6–10, 2023.

15 TASCA, D. *et al.* A 2.9–4.0-ghz fractional-n digital pll with bang-bang phase detector and 560-fs_{rms} integrated jitter at 4.5-mw power. **IEEE Journal of Solid-State Circuits**, v. 46, n. 12, p. 2745–2758, 2011.

16 LIU, H.-E. *et al.* A low-power spread spectrum clock generator with an embeddable half-integer division ratio interpolator. *In*: **2014 IEEE International Symposium on Circuits and Systems (ISCAS)**. [*S.l.: s.n.*], 2014. p. 1873–1876.

17 LIN, J.; JIANG, H.; CHI, B. A 5-bit phase-interpolator-based fractional-n frequency divider for digital phase-locked loops. *In*: **2017 IEEE International Symposium on Circuits and Systems (ISCAS)**. [*S.l.: s.n.*], 2017. p. 1–4.

18 XU, H. *et al.* A flexible 0.73–15.5 ghz single lc vco clock generator in 12 nm cmos. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 69, n. 11, p. 4238–4242, 2022.

19 HUANG, H.-Y.; HO, S.-F.; SU, H.-Y. A dual-modulus prescaler using double-edge-triggered d-flip-flops. *In*: **2003 46th Midwest Symposium on Circuits and Systems**. [*S.l.: s.n.*], 2003. v. 1, p. 209–212 Vol. 1.

20 MOON, Y.; LEE, S.-H.; SHIM, D. A divide-by-16.5 circuit for 10-gb ethernet transceiver in 0.13-/spl mu/m cmos. **IEEE Journal of Solid-State Circuits**, v. 40, n. 5, p. 1175–1179, 2005.

21 LU, L. *et al.* A 975-to-1960mhz fast-locking fractional-n synthesizer with adaptive bandwidth control and 4/4.5 prescaler for digital tv tuners. *In*: **2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers**. [*S.l.: s.n.*], 2009. p. 396–397,397a.

22 JI-REN, Y.; KARLSSON, I.; SVENSSON, C. A true single-phase-clock dynamic cmos circuit technique. **IEEE Journal of Solid-State Circuits**, v. 22, n. 5, p. 899–901, 1987.

23 SUZUKI, Y.; ODAGAWA, K.; ABE, T. Clocked cmos calculator circuitry. **IEEE Journal of Solid-State Circuits**, v. 8, n. 6, p. 462–469, 1973.

24 KRAMBECK, R.; LEE, C.; LAW, H.-F. High-speed compact circuits with cmos. **IEEE Journal of Solid-State Circuits**, v. 17, n. 3, p. 614–619, 1982.

25 GONCALVES, N.; MAN, H. D. Nora: a racefree dynamic cmos technique for pipelined logic structures. **IEEE Journal of Solid-State Circuits**, v. 18, n. 3, p. 261–266, 1983.

26 YUAN, J.; SVENSSON, C. High-speed cmos circuit technique. **IEEE Journal of Solid-State Circuits**, v. 24, n. 1, p. 62–70, 1989.

27 YUAN, J.; SVENSSON, C. New single-clock cmos latches and flipflops with improved speed and power savings. **IEEE Journal of Solid-State Circuits**, v. 32, n. 1, p. 62–69, 1997.

28 RAZAVI, B. Tspc logic [a circuit for all seasons]. **IEEE Solid-State Circuits Magazine**, v. 8, n. 4, p. 10–13, 2016.

29 NAVARRO, J. **Técnicas para projetos de ASICs CMOS de alta velocidade**. 1998. Tese (PhD thesis) — University of São Paulo, São Paulo, BR, 1998.

30 S., J. N.; NOIJE, W. A. M. V. E-tspc: Extended true single-phase-clock cmos circuit technique. *In*: _____. **VLSI: Integrated Systems on Silicon: IFIP TC10 WG10.5 International Conference on Very Large Scale Integration 26–30 August 1997, Gramado, RS, Brazil**. Boston, MA: Springer US, 1997. p. 165–176. ISBN 978-0-387-35311-1. Available at: https://doi.org/10.1007/978-0-387-35311-1_14.

31 AFGHAHI, M.; YUAN, J. Double-edge-triggered d-flip-flops for high-speed cmos circuits. **IEEE Journal of Solid-State Circuits**, v. 26, n. 8, p. 1168–1170, 1991.

32 NAVARRO, J.; LUPPE, M. Performance comparison of high-speed dual modulus prescalers using metaheuristic sizing/optimization. *In*: **2020 33rd Symposium on Integrated Circuits and Systems Design (SBCCI)**. [*S.l.: s.n.*], 2020. p. 1–6.

33 WEBER, T. O.; NOIJE, W. A. M. V. Design of analog integrated circuits using simulated annealing/quenching with crossovers and particle swarm optimization. *In*: TSUZUKI, M. de S. G. (ed.). **Simulated Annealing**. Rijeka: IntechOpen, 2012. cap. 11. Available at: https://doi.org/10.5772/50384.

34 KENNEDY, J.; EBERHART, R. Particle swarm optimization. *In*: **Proceedings of ICNN'95 - International Conference on Neural Networks**. [*S.l.: s.n.*], 1995. v. 4, p. 1942–1948 vol.4.

35 CLERC, M.; KENNEDY, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 1, p. 58–73, 2002.

36 KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, v. 220, n. 4598, p. 671–680, 1983. Available at: https://www.science.org/doi/abs/10.1126/science.220.4598.671.

APPENDIX A – OPTIMIZATION VARIABLES

The MOSFET parameters width and length were configured as variables for the metaheuristics-based optimization framework. Additional information regarding these variables and their respective search ranges is presented in the following tables:

Table A1 lists the variables used for optimizing the **DETFF** prescaler and their associations with the physical parameters of the circuit shown in Figure 14. The channel length of all transistors was set to 65 nm.

Table A2 lists the variables used for optimizing the **DDR** prescaler and their associations with the physical parameters of the circuit shown in Figure 17. The channel length of all transistors not indicated in Table A2 was set to 65 nm.

Table A3 lists the variables used for optimizing the **B-DDRv1** prescaler and their associations with the physical parameters of the circuit shown in Figure 20. The channel length of all transistors not indicated in Table A3 was set to 65 nm.

Table A4 lists the variables used for optimizing the **B-DDRv2** prescaler and their associations with the physical parameters of the circuit shown in Figure 21. The channel length of all transistors not indicated in Table A4 was set to 65 nm.

Table A1 – DETFF Prescaler Optimization Variables.
(W_{XX} represents the width of the transistor M_{XX})

Variable	Physical Parameter		Range	
			Start	End
X1	W_{P1}, W_{P9}	(µm)	0.18	5.0
X2	W_{N1}, W_{N9}	(µm)	0.18	5.0
X3	W_{P2}, W_{P10}	(µm)	0.18	5.0
X4	W_{N2}, W_{N10}	(µm)	0.18	5.0
X5	W_{P3}, W_{P11}	(µm)	0.18	5.0
X6	W_{N3}, W_{N11}	(µm)	0.18	5.0
X7	W_{P4}, W_{P12}	(µm)	0.18	5.0
X8	W_{N4}, W_{N12}	(µm)	0.18	4.0
X9	W_{P5}, W_{P13}	(µm)	0.18	5.0
X10	W_{N5}, W_{N13}	(µm)	0.18	4.0
X11	W_{P6}, W_{P14}	(µm)	0.18	4.0
X12	W_{N6}, W_{N14}	(µm)	0.18	4.0
X13	W_{P7}	(µm)	0.18	4.0
X14	W_{N7}	(µm)	0.18	4.0
X15	W_{P8}	(µm)	0.18	4.0
X16	W_{P8}	(µm)	0.18	4.0
X17*	W_{P15} / 2.0	(µm)	0.2	0.8

* (W_{P15} + W_{N15} = 2.0 µm)

Table A2 – DDR Prescaler Optimization Variables.
(W_{XX} and L_{XX} represent the width and length of the transistor M_{XX}, respectively)

Variable	**Physical Parameter**		**Range**	
			Start	**End**
X1	L_{P1}, L_{P5}	(×65 nm)	1	10
X2	W_{P1}, W_{P5}	(µm)	0.18	4.0
X3	W_{N1}, W_{N7}	(µm)	0.18	4.0
X4	W_{N2}, W_{N8}	(µm)	0.18	4.0
X5	W_{N3}, W_{N9}	(µm)	0.18	4.0
X6	W_{P2}, W_{P6}	(µm)	0.18	4.0
X7	W_{N4}, W_{N10}	(um)	0.18	4.0
X8	L_{N5}, L_{N11}, L_{N12}	(×65 nm)	1	10
X9	W_{P3}, W_{P7}	(µm)	0.18	4.0
X10	W_{N5}, W_{N11}	(µm)	0.18	4.0
X11	W_{P8}	(µm)	0.18	4.0
X12	W_{N12}	(µm)	0.18	4.0
X13	W_{P4}	(µm)	0.18	4.0
X14	W_{N6}	(µm)	0.18	4.0
X15	L_{P9}	(×65 nm)	1	10
X16	W_{P9}	(µm)	0.18	4.0
X17	W_{N13}	(µm)	0.18	4.0
X18	W_{P11}	(µm)	0.18	4.0
X19	W_{P10}	(µm)	0.18	4.0
X20	W_{N14}	(µm)	0.18	4.0
X21*	W_{P12} / 2.0	(µm)	0.2	0.8

* (W_{P12} + W_{N15} = 2.0 µm)

Table A3 – B-DDRv1 Prescaler Optimization Variables.
(W_{XX} and L_{XX} represent the width and length of the transistor M_{XX}, respectively)

Variable	**Physical Parameter**		**Range**	
			Start	**End**
X1	L_{P1}, L_{P4}	(×65 nm)	1	10
X2	W_{P1}, W_{P4}	(µm)	0.18	4.0
X3	W_{N1}, W_{N5}	(µm)	0.18	4.0
X4	W_{N2}, W_{N6}	(µm)	0.18	4.0
X5	W_{P2}, W_{P5}	(µm)	0.18	4.0
X6	L_{N3}, L_{N7}	(×65 nm)	1	10
X7	W_{N3}, W_{N7}	(µm)	0.18	4.0
X8	W_{P3}, W_{P6}	(µm)	0.18	4.0
X9	L_{N4}, L_{N8}	(×65 nm)	1	10
X10	W_{N4}, W_{N8}	(µm)	0.18	4.0
X11	W_{P7}, W_{P8}	(µm)	0.18	4.0
X12	W_{N9}, W_{N10}	(µm)	0.18	4.0
X13*	W_{P9} / 2.0	(µm)	0.2	0.8

* (W_{P9} + W_{N11} = 2.0 µm)

Table A4 – B-DDRv2 Prescaler Optimization Variables.
(W_{XX} and L_{XX} represent the width and length of the transistor M_{XX}, respectively)

Variable	**Physical Parameter**		**Range**	
			Start	**End**
X1	L_{P1}, L_{P4}	(×65 nm)	1	10
X2	W_{P1}, W_{P4}	(µm)	0.18	4.0
X3	W_{N1}, W_{N5}	(µm)	0.18	4.0
X4	W_{N2}, W_{N6}	(µm)	0.18	4.0
X5	W_{P2}, W_{P5}	(µm)	0.18	4.0
X6	L_{N3}, L_{N7}	(×65 nm)	1	10
X7	W_{N3}, W_{N7}	(µm)	0.18	4.0
X8	W_{P3}, W_{P6}	(µm)	0.18	4.0
X9	W_{N4}, W_{N8}	(µm)	0.18	4.0
X10	W_{P7}, W_{P8}	(µm)	0.18	4.0
X11	W_{N9}, W_{N10}	(µm)	0.18	4.0
X12*	W_{P9} / 2.0	(µm)	0.2	0.8

* (W_{P9} + W_{N11} = 2.0 µm)

APPENDIX B – PRELIMINARY HIGH-SPEED DATA

Figure B1 presents preliminary data showing the average power consumption versus operating frequency for the high-speed operation of the DDR, B-DDRv1, and B-DDRv2 designs. Both the DDR and B-DDRv1 were optimized for operation at 13.5 GHz, while the B-DDRv2 was optimized for a 16.5 GHz input clock.

Figure B1 – High-Speed Power Consumption versus Frequency.
(DDR and B-DDRv1 optimized for 13.5 GHz, B-DDRv2 optimized for 16.5 GHz)